

# **Computational Job Market Analysis**

## **with Natural Language Processing**

Mike Zhang  
张军军

This thesis has been submitted to the Ph.D. School of the  
IT University of Copenhagen on 14 January 2024



The research for this doctoral thesis has received funding from the Independent Research Fund Denmark (DFF) grant 9131-00019B and in parts by European Research Council (ERC) Consolidator grant DIALECT 101043235.





## **Committee**

**Advisor**      prof. dr. B. Plank      Ludwig-Maximilians-Universität München  
IT Universitetet i København

**Co-Advisor**    dr. R.M. van der Goot    IT Universitetet i København

**Members**      prof. dr. C. Biemann      Universität Hamburg  
dr. T. Bogers (chair)    IT Universitetet i København  
dr. D. Maynard          University of Sheffield



## Abstract

Recent technological advances underscore the dynamic nature of the labor market. These transformative shifts yield significant consequences for employment prospects, resulting in the increase of job vacancy data across platforms and languages. The aggregation of such data holds the potential to gain valuable insights into labor market demands, the emergence of new skills, and the overall facilitation of job matching. These benefits extend to various parties, including job platforms, recruitment agencies, applicants, and other stakeholders within the ecosystem. However, despite the prevalence of such insights in the private sector, we lack transparent language technology systems and data for this domain.

The primary objective of this thesis is to investigate the use of Natural Language Processing (NLP) technology for the extraction of relevant information from job descriptions. We identify several general challenges within this domain. These encompass a scarcity of available training and evaluation data, a lack of standardized guidelines to annotate data, and a shortage of effective methods for extracting information from job ads.

Therefore, we embark on a comprehensive study of the entire process: First, framing the problem and getting annotated data for training NLP models. Here, our contributions encompass job description datasets, including a de-identification dataset, and a novel active learning algorithm designed for efficient model training. Second, we introduce several extraction methodologies to tackle the task of information extraction from job advertisement data: A skill extraction approach using weak supervision, a taxonomy-aware pre-training methodology adapting a multilingual language model to the job market domain, and a retrieval-augmented model leveraging multiple skill extraction datasets to enhance overall extraction performance. Lastly, given the extracted information, we delve into the grounding of this data within a designated taxonomy.

## Resumé

De seneste teknologiske fremskridt understreger arbejdsmarkedets dynamiske natur. Disse transformative skift har betydelige konsekvenser for beskæftigelsesudsigterne, hvilket resulterer i en stigning i data om ledige stillinger på tværs af platforme og sprog. Aggregering af sådan data har potentiale til at give værdifuld indsigt i arbejdsmarkedets efterspørgsler, fremkomst af nye færdigheder, og general facilitering af jobmatching. Disse fordele kommer forskellige parter til gode, herunder jobplatforme, rekrutteringsbureauer, ansøgere og andre interessenter i økosystemet. På trods af udbredelsen af sådanne indsigter i den private sektor, mangler vi gennemsigtige sprogteknologiske systemer og data til dette domæne.

Det primære formål med denne afhandling er at undersøge brugen af sprogteknologi (*en*: Natural Language Processing) til at udtrække relevant information fra jobbeskrivelser. Vi identificerer flere generelle udfordringer inden for dette domæne. Disse omfatter en mangel på tilgængelig trænings- og evalueringsdata, en mangel på standardiserede retningslinjer for annotering af data samt en mangel på effektive metoder til at udtrække information fra jobannoncer.

Derfor går vi i gang med en omfattende undersøgelse af hele processen: Indledningsvis formuleres problemet og data annoteres med henblik på træning af NLP-modeller. Vores bidrag omfatter datasæt med jobbeskrivelser, herunder et de-identifikationsdatasæt, og en ny aktiv læringsalgoritme designet til effektiv modeltræning. For det andet introducerer vi flere ekstraktionsmetoder til at tackle opgaven med at udtrække information fra jobannoncedata: En metode til udtrækning af færdigheder ved hjælp af svag supervision, en taksonomi-bevidst prætræningsmetode, der tilpasser en flersproget sprogmodel til jobmarkedsdomænet, samt en informationssøgning-baseret model, der udnytter flere datasæt til at udtrække færdigheder for at forbedre den samlede effektivitet. Til sidst, givet den ekstraherede information dykker vi ned i forankringen af disse data inden for en udpeget taksonomi.

## Acknowledgements

PhDone! It's with immense joy and gratitude that I share my thesis, a testament to the journey of the past three years. This ride has been a wild one—filled with highs, lows, and everything in between. Along the way, I've crossed paths with countless incredible souls who have left an indelible mark on my heart, and it's only fitting to express my deepest thanks to each and every one of them.

First and foremost, Barbara and Rob, your unwavering support and belief in me have been the bedrock upon which this entire endeavor was built. Without your guidance and encouragement, this journey would have been a daunting one indeed.

To the Groningen NLP crew, especially Antonio, you all lit the spark of curiosity within me and showed me the magic of research. Your passion was infectious, and our collaboration was nothing short of inspiring. To my fellow masters' cohorts—Leon, Gerben, Roy, Wessel, Ian, Léon, Aria, Yu-Wen—thank you for the camaraderie and the brainstorming sessions for projects and such that made it all worthwhile, and also for making me interested in pursuing a Ph.D.

Thanks to the folks at Dataprovider for supporting me in leaving the industry and going to academia: Wouter, Maaïke, Edine, and the rest of the folks, thank you. Andriana, your keen eye and knack for words turned my applications into polished gems, and I'm forever grateful for your expertise.

Starting a PhD amid a pandemic was certainly a curveball, but with the unwavering support of the research group, we weathered the storm together, emerging stronger and more resilient. In the end, thank you Kristian, Dennis, Daniel, David, Joris, Maria, Marija, Peter, Christian, Stefan, Anna, Meisam, and Anders (special thanks for helping me with the Danish version of my abstract, tak skal du have).

During my Ph.D., I was fortunate to do a small internship at NEC Laboratories Europe, which gave me fresh insights into industry research, thanks to Carolin and Kiril for making it possible. To my cohort—Gorjan, Marcello, Iris, Parul, Foad, Huixu—you made it much more fun in Heidelberg.

To my squad back home—Luuk, Althia, Bram, Lieve, Dion, Nathalie, Sietse, Mirjam, Pim, Bert, Jasper (may you rest in peace), Huub, Ruben, Frank, Martijn, Bas, Tjalling, Britt, Kevin, Yihang, Shawn, Sheng—your friendship has been my anchor through the choppy waters of academia.

And to my Copenhagen friends—Alex, David, and Maria Paula—thanks for keeping me sane with your laughter and camaraderie.

Special shoutout to Elisa (grazie) and Max (ありがとう), my partners in crime in the office and dear friends, I owe you both a debt of gratitude beyond measure. Max, your endless knowledge and knack for finding humor helped make the office great fun. And Elisa, your kindness and unwavering support have been a constant source of strength for me. For you both, whether it was bouncing ideas off each other, sharing laughs over lunch, or simply being there to listen, I am endlessly grateful for your friendship.

Last but certainly not least, to my family—Tante Lin, Nick, Tante Lin Hong, Nancy, Milan, Mandy, Reyno, Andy, Oom Lin Yang, Mama, Papa, and Fify—your unwavering love and support have been quintessential in finishing up the Ph.D.

To everyone who has touched my life in one way or another, thank you. This journey wouldn't have been half as meaningful without you all by my side.

–Mike

Copenhagen, March 2024

## **Declaration of Work**

I, Mike Zhang, declare that this thesis - submitted in partial fulfillment of the requirements for the conferral of PhD, from the IT University of Copenhagen - is solely my own work unless otherwise referenced or attributed. Neither the thesis nor its content have been submitted (or published) for qualifications at another academic institution.

– Mike Zhang

# Table of Contents

Abstract . . . . .	v
Resumé . . . . .	vi
Acknowledgements . . . . .	vii
Declaration of Work . . . . .	ix
<b>1 Introduction</b>	<b>1</b>
1.1 Annotating Data (Part II) . . . . .	2
1.2 Modeling Occupational Skills (Part III) . . . . .	7
1.3 Linking Skills to Existing Resources (Part IV) . . . . .	9
1.4 Chapter Guide & Contributions . . . . .	10
1.5 Publications . . . . .	12
<b>I Background</b>	<b>17</b>
<b>2 Challenges with Skill Extraction in the Job Market Domain</b>	<b>19</b>
2.1 Defining Skills . . . . .	19
2.1.1 The European Skills, Competences, Qualifications and Occupations . . . . .	19
2.1.2 Challenges . . . . .	22
2.2 Skill Extraction . . . . .	25
2.2.1 Skill Extraction Definition . . . . .	25
2.2.2 Span Extraction with Neural Networks . . . . .	26
2.2.3 Evaluation . . . . .	26
2.3 Categorization of Skills . . . . .	28
2.3.1 Coarse-grained Skill Categorization . . . . .	28



2.3.2	Fine-grained Skill Categorization . . . . .	29
2.4	Annotating Data . . . . .	30
2.4.1	Personally Identifiable Information . . . . .	30
2.4.2	Active Learning . . . . .	31
2.4.3	Weak Supervision . . . . .	32
2.5	Analysis . . . . .	32
2.5.1	Span Length . . . . .	33
2.5.2	Long-tail Skills . . . . .	33
2.5.3	False Positives . . . . .	34
2.6	Transfer to the Job Market Domain . . . . .	35
2.6.1	Motivation . . . . .	35
2.6.2	Multi-task Learning . . . . .	35
2.6.3	Transfer Learning . . . . .	36
<b>II</b>	<b>Annotating Data</b>	<b>39</b>
<b>3</b>	<b>De-identification of Privacy Related Entities in Job Postings</b>	<b>41</b>
3.1	Introduction . . . . .	42
3.2	Related Work . . . . .	43
3.2.1	De-identification in the Medical Domain . . . . .	43
3.2.2	Rule-based . . . . .	44
3.2.3	Traditional ML . . . . .	44
3.2.4	Neural methods . . . . .	45
3.2.5	De-identification in other Domains . . . . .	45
3.3	JOBSTACK . . . . .	46
3.3.1	Statistics . . . . .	48
3.3.2	Annotation Schema . . . . .	49
3.3.3	Annotation Quality . . . . .	50
3.4	Methods . . . . .	51
3.4.1	Models . . . . .	51
3.4.2	Embeddings . . . . .	52
3.4.3	Auxiliary tasks . . . . .	52

3.5	Evaluation . . . . .	54
3.5.1	Is a CRF layer necessary? . . . . .	55
3.5.2	LSTM versus Transformer . . . . .	55
3.5.3	Poor performance with BERT <sub>Overflow</sub> . . . . .	55
3.5.4	Auxiliary data increases performance . . . . .	56
3.5.5	Performance on the test set . . . . .	57
3.5.6	Per-entity Analysis . . . . .	57
3.6	Conclusions . . . . .	58
3.7	Appendix . . . . .	60
3.7.1	Example Job Posting . . . . .	60
<b>4</b>	<b>Cartography Active Learning</b>	<b>61</b>
4.1	Introduction . . . . .	62
4.2	Related Work . . . . .	63
4.2.1	Common Strategies . . . . .	64
4.2.2	Batch-mode Active Learning . . . . .	64
4.2.3	Batch-aware Query Strategies . . . . .	65
4.3	Cartography Active Learning . . . . .	66
4.3.1	Mapping the Data . . . . .	67
4.3.2	Data Maps from Seed Data . . . . .	68
4.3.3	CAL Algorithm . . . . .	69
4.4	Experimental Setup . . . . .	70
4.4.1	Datasets . . . . .	70
4.4.2	Acquisition Functions . . . . .	71
4.4.3	Configurations . . . . .	73
4.5	Results & Analysis . . . . .	75
4.5.1	Why does CAL work? . . . . .	75
4.5.2	What is the influence of the <i>correctness</i> threshold? . . . . .	77
4.5.3	Do AL strategies select the same instances for labeling? . . . . .	77
4.5.4	How data-efficient is CAL in comparison to full data training? . . . . .	78
4.6	Conclusion . . . . .	79
4.7	Appendix . . . . .	80

4.7.1	Reproducibility . . . . .	80
4.7.2	Full Data Map . . . . .	80
<b>5</b>	<b>SKILLSPAN: Hard and Soft Skill Extraction from English Job Postings</b>	<b>83</b>
5.1	Introduction . . . . .	84
5.2	Related Work . . . . .	88
5.3	Skill & Knowledge Definition . . . . .	89
5.4	SKILLSPAN Dataset . . . . .	92
5.4.1	Data . . . . .	92
5.4.2	Data Annotation . . . . .	93
5.4.3	Further Details on the Annotation Process . . . . .	93
5.4.4	Annotation Span Statistics . . . . .	95
5.4.5	Qualitative Analysis of Annotations . . . . .	96
5.5	Experimental Setup . . . . .	96
5.5.1	BERT <sub>base</sub> (Devlin et al., 2019b) . . . . .	97
5.5.2	SpanBERT (Joshi et al., 2020b) . . . . .	97
5.5.3	JobBERT . . . . .	97
5.5.4	JobSpanBERT . . . . .	98
5.5.5	Experiments . . . . .	98
5.5.6	Setup . . . . .	98
5.6	Results . . . . .	100
5.6.1	Performance on Development Set . . . . .	100
5.6.2	Performance on Test Set . . . . .	100
5.6.3	Significance . . . . .	101
5.7	Discussion . . . . .	102
5.7.1	What Did Not Work . . . . .	102
5.7.2	Continuous Pretraining helps SE . . . . .	102
5.7.3	Difference in Length of Predictions . . . . .	103
5.7.4	Performance per Span Length . . . . .	105
5.8	Conclusion . . . . .	105
5.9	Appendix . . . . .	107
5.9.1	Data Statement SKILLSPAN . . . . .	107

5.9.2	Annotation Guidelines . . . . .	108
5.9.3	Knowledge Specifications . . . . .	111
5.9.4	Other Specifications . . . . .	113
5.9.5	Type of Skills Annotated . . . . .	115
5.9.6	Reproducibility . . . . .	115
5.9.7	Exact Number of Performance . . . . .	115
5.9.8	Significance Testing . . . . .	117

## **6 Kompetencer: Fine-grained Skill Classification in Danish Job**

	<b>Postings via Distant Supervision and Transfer Learning</b>	<b>123</b>
6.1	Introduction . . . . .	124
6.2	KOMPETENCER Dataset . . . . .	126
6.2.1	Skill & Knowledge Definition . . . . .	126
6.2.2	Dataset Statistics . . . . .	128
6.2.3	Annotations . . . . .	130
6.3	Methodology . . . . .	132
6.3.1	Encoders . . . . .	133
6.3.2	Multilingual Encoder . . . . .	133
6.3.3	Experimental Setup . . . . .	133
6.4	Analysis of Results . . . . .	134
6.4.1	Performance Zero-shot Setting . . . . .	134
6.4.2	Performance Few-shot Setting . . . . .	136
6.4.3	Combining Training Data . . . . .	136
6.4.4	Is Domain Adaptive Pretraining Worth It? . . . . .	136
6.4.5	Analysis of Predictions . . . . .	138
6.4.6	Qualitative Analysis Distant Supervision . . . . .	138
6.5	Related Work . . . . .	138
6.6	Conclusion . . . . .	139
6.7	Appendix . . . . .	141
6.7.1	Data Statement KOMPETENCER . . . . .	141
6.7.2	Reproducibility . . . . .	142
6.7.3	Label Meaning . . . . .	142
6.7.4	Exact Results from Plots . . . . .	142

6.7.5	Confusion Matrix Few-Shot . . . . .	143
<b>III</b>	<b>Modeling Occupational Skills</b>	<b>149</b>
<b>7</b>	<b>Skill Extraction from Job Postings using Weak Supervision</b>	<b>151</b>
7.1	Introduction . . . . .	152
7.2	Methodology . . . . .	153
7.2.1	Data . . . . .	154
7.2.2	ESCO Statistics . . . . .	155
7.2.3	Baselines . . . . .	156
7.2.4	Skill Representations . . . . .	157
7.2.5	Matching . . . . .	158
7.3	Analysis of Results . . . . .	159
7.3.1	Results . . . . .	159
7.3.2	Qualitative Analysis . . . . .	161
7.4	Conclusion . . . . .	161
7.5	Appendix . . . . .	162
7.5.1	Exact Results . . . . .	164
7.5.2	Definition F1 . . . . .	164
7.5.3	Exact Numbers Results . . . . .	164
<b>8</b>	<b>ESCOXLM-R: Multilingual Taxonomy-driven Pre-training for the Job Market Domain</b>	<b>165</b>
8.1	Introduction . . . . .	166
8.2	ESCOXLM-R . . . . .	169
8.2.1	Preliminaries . . . . .	169
8.2.2	Multilinguality . . . . .	169
8.2.3	European Skills, Competences, Qualifications and Occupations Taxonomy . . . . .	169
8.2.4	Pre-training Setup . . . . .	172
8.2.5	Pre-training Objectives . . . . .	173
8.2.6	Implementation . . . . .	174
8.3	Experimental Setup . . . . .	175

8.3.1	SKILLSPAN (Zhang et al., 2022a)	175
8.3.2	KOMPETENCER (Zhang et al., 2022c)	175
8.3.3	BHOLA (Bhola et al., 2020)	176
8.3.4	SAYFULLINA (Sayfullina et al., 2018)	176
8.3.5	GREEN (Green et al., 2022)	176
8.3.6	JOBSTACK (Jensen et al., 2021b)	177
8.3.7	GNEHM (Gnehm et al., 2022b)	177
8.3.8	FIJO (Beauchemin et al., 2022)	177
8.4	Results	177
8.4.1	English	178
8.4.2	Non-English	178
8.4.3	Analysis	179
8.5	Discussion	179
8.5.1	Performance on Span Length	179
8.5.2	Entity-F1 vs. Surface-F1	181
8.6	Related Work	182
8.7	Conclusion	183
8.8	Appendix	186
8.8.1	Example Extraction from ESCO	186
8.8.2	Data Examples	188
8.8.3	Fine-tuning Details	195
<b>9</b>	<b>NNOSE: Nearest Neighbor Occupational Skill Extraction</b>	<b>197</b>
9.1	Introduction	198
9.2	Nearest Neighbor Skill Extraction	200
9.2.1	Skill Extraction.	200
9.2.2	NNOSE	201
9.2.3	Datastore	201
9.2.4	Inference	201
9.2.5	Whitening Transformation	202
9.3	Experimental Setup	203
9.3.1	Data	203
9.3.2	Models	204

9.4	Results	204
9.4.1	Best Model Performance	206
9.4.2	Best NNOSE Setting	206
9.5	Analysis	207
9.5.1	Long-tail Skills Prediction	207
9.5.2	Retrieving From All Datasets	208
9.5.3	Prediction of Unseen Skills	209
9.5.4	Cross-dataset Long-tail Analysis	211
9.5.5	Qualitative Check on Prediction Errors	212
9.6	Related Work	213
9.6.1	Skill Extraction	213
9.6.2	General Retrieval-augmentation	213
9.6.3	Decoder-based Nearest Neighbor Approaches	214
9.6.4	Encoder-based Nearest Neighbor Approaches	214
9.7	Conclusion	214
9.8	Appendix	217
9.8.1	Whitening Transformation Algorithm	217
9.8.2	Data Examples	219
9.8.3	Implementation Details	219
9.8.4	General Implementation	219
9.8.5	JobBERTa	220
9.8.6	NNOSE Setup	221
9.8.7	Hyperparameters NNOSE	222
9.8.8	Development Set Results	223
9.8.9	Frequency Distribution of Skills	223
9.8.10	Qualitative Results NNOSE	223
9.9	Further Cross-dataset Analysis	225

## **IV Linking Skills to Existing Sources 229**

### **10 Entity Linking in the Job Market Domain 231**

10.1	Introduction	232
------	--------------	-----

10.2 Methodology . . . . .	234
10.2.1 Definition . . . . .	234
10.2.2 Data . . . . .	234
10.2.3 Models . . . . .	235
10.2.4 Setup . . . . .	237
10.3 Results . . . . .	238
10.4 Discussion . . . . .	239
10.4.1 Qualitative Analysis . . . . .	239
10.4.2 Evaluation Limitation . . . . .	240
10.5 Conclusion . . . . .	240
10.6 Appendix . . . . .	242
10.6.1 Data Preprocessing . . . . .	242
10.6.2 Data Examples . . . . .	243
10.6.3 Implementation Details . . . . .	244
10.6.4 UNK Evaluation . . . . .	244
<b>V Conclusions</b>	<b>247</b>
<b>11 Summary and Conclusions</b>	<b>249</b>
11.1 Future Directions . . . . .	255
<b>Bibliography</b>	<b>257</b>



# Chapter 1

## Introduction

From the 2010s until the 2020s, we were experiencing the Fourth Industrial Revolution ([Lasi et al., 2014](#); [Schwab, 2017](#)), characterized by a digital revolution and more mobile internet. The cost of the technologies used is also decreasing as they develop. At the time of writing, 1TB of disk storage costs around 14 EUR, whereas when the author of this thesis was born, the price was 128,000 EUR. Such low-cost access to technologies drove innovation. Reasonably affordable graphical processing units (GPUs) were one of the reasons for the advances in Artificial Intelligence (AI) and Deep Learning (DL; [Krizhevsky et al., 2012](#); [LeCun et al., 2015](#); [Goodfellow et al., 2016](#)). Since 2021, AI has been seen as one of the main drivers for Industry 5.0 ([European Commission, 2021](#)). Such adoptions of technological changes have implications for the ever-changing labor market ([Brynjolfsson and McAfee, 2011, 2014](#)). One concrete example is Large Language Models (LLMs) and especially ChatGPT ([OpenAI, 2023](#)) in December 2022.<sup>1</sup> Investigations show the impact of the technology on the labor market ([Eloundou et al., 2023](#)):

“80% of the U.S. workforce could have at least 10% of their work tasks affected by LLMs and 19% of the U.S. workforce

---

<sup>1</sup>We would like to note that most of the work done in this thesis was before the emergence of LLMs in December 2022.

may see at least 50% of their work tasks affected by LLMs.”

Therefore, we can expect a surge of job vacancies emerging on a variety of platforms and in big quantities. Investigating these job vacancies can provide insights into labor market dynamics, such as skill demands or job matching (Holt and David, 1966; Balog et al., 2012; Azar et al., 2022). To infer occupational skill demands automatically, one can apply natural language processing (NLP) technology to the job vacancy data to extract relevant skills from the job ad, e.g., “Python programming” or “communication skills”. Once these skills are extracted, we could deduce skill demand and get a snapshot of the labor market.

In this thesis, we investigate how we can leverage NLP to extract relevant information from job vacancy data, and in part how to further standardize the extracted information. More specifically, in this thesis, we concern ourselves with the task of Skill Extraction (SE): The task of extracting relevant spans from a text that resembles a skill. SE is one of the core applications within a field we depict as *Computational Job Market Analysis*.

The bottleneck of this task is multi-fold, we refer to these challenges in [Subsection 2.1.2](#). There are also other general challenges surrounding this task, not only limited to SE, such as data annotation and how we can standardize these extracted skills. The necessary background for this domain is presented in [Chapter 2 \(Part I\)](#). In the following sections, we introduce several research questions, which are also the larger overall themes of this thesis ([Part II](#), [III](#), and [IV](#)).

## 1.1 Annotating Data (Part II)

Job postings can contain privacy-related entities, such as phone numbers, names, and other personally identifiable information. To comply with, e.g., the European General Data Protection Regulation (GDPR; [European Commission, 2016](#)), we need to remove such information from the text before we continue extracting other types of information (e.g.,

skills). Manually masking or removing this information is time-consuming and labor-intensive. To the best of our knowledge, there exists no de-identification work in the job posting domain. There are several works on de-identifying parts in other sources of text, such as the medical domain (e.g., Szarvas et al., 2007; Meystre et al., 2010; Liu et al., 2015; Jiang et al., 2017; Friedrich et al., 2019; Trienes et al., 2020) or SMS messages (e.g., Treurniet et al., 2012; Patel et al., 2013; Chen and Kan, 2013). The task at the time was frequently approached with a bidirectional Long-Short Term Memory (Hochreiter et al., 1997) model (BiLSTM). We explore how Transformer-based models compare against recurrent neural networks (e.g., BiLSTMs).

Additionally, as there are de-identification datasets available in other domains, we investigate how a model can benefit by learning from these other auxiliary, related tasks via multi-task learning (Caruana, 1997). In this thesis, we annotate for these privacy-related entities in job postings and seek to answer the following research questions (RQs):

---

### Chapter 3

**RQ1** How do Transformer-based models compare against recurrent neural network models for de-identification of privacy-related entities in job postings?

**RQ2** Which auxiliary tasks and datasets improve de-identification performance of privacy-related entities from job postings?

---

Generally, annotation is a labor-intensive and expensive process. A potential remedy involves the use of active learning (Cohn et al., 1994; Lewis and Catlett, 1994; Settles, 2009). The fundamental concept behind active learning is that a Machine Learning (ML) model can enhance its accuracy

using fewer labeled training instances when it can *select* the data for learning. This approach is extensively used to address the challenges associated with the time-consuming and costly tasks of manually labeling data.

The most straightforward and widely employed approach is *uncertainty sampling* (Lewis and Gale, 1994; Lewis and Catlett, 1994), wherein the learner queries instances about which it is most uncertain. However, uncertainty sampling is myopic, as it solely measures the information content of an individual data instance. Alternative AL algorithms focus on assembling a *diverse* batch for annotation (Geifman and El-Yaniv, 2017; Sener and Savarese, 2018; Gissin and Shalev-Shwartz, 2019; Zhdanov, 2019) or estimating the *uncertainty* distribution of the model (Houlsby et al., 2011; Gal and Ghahramani, 2016). Nevertheless, these methods are typically constrained in their definition of informativeness, which is associated with post-training model uncertainty and batch diversity. Instead, it could be more useful if the behavior of the model on individual instances during training could be exploited, which is usually called training dynamics (Swayamdipta et al., 2020b), we investigate:

---

## Chapter 4

**RQ3** Can training dynamic-based active learning lead to more efficient model training than previous methods for text classification while maintaining performance?<sup>2</sup>

---

In the context of SE, annotation guidelines are lacking, it is unclear when a skill span starts and when it ends. Consequentially, there exists a lacuna of high-quality annotated data to train ML models for this domain. There

---

<sup>2</sup>Here, we do not apply it to the job market domain or in the context of sequence labeling, but instead to text classification tasks in general domains (e.g., news and question types).

are, however, existing alternative resources such as the European Skills, Competences, Qualifications, and Occupations (ESCO; [le Vrang et al., 2014](#)). This is a large-scale taxonomy consisting of textual descriptions of occupations and skills (3K and 14K respectively). For the skills part of the taxonomy, there are descriptions and other possible denominations. For example, a skill is “plan teamwork” and an alternative to this is “plan to work as a team”.<sup>3</sup> However, these lists of alternative names for skills are non-exhaustive and therefore, are hard to match directly to a job posting. Additional reasons, for example, are not having a string match, due to small variations of wording (e.g., driver’s license vs. driving license). There could also be misspellings. Therefore, we instead develop our annotation guidelines with inspiration from the definitions and spans in ESCO ([Chapter 5](#)).

Alternatively, existing resources are also the unannotated job postings themselves. Previous work has shown that further training language models (domain-adaptive pre-training or continuous pre-training; [Gururangan et al., 2020b](#)) on large corpora of a specific domain (e.g., the medical domain) shows a significant performance improvement on the downstream tasks of that domain. We explore two questions:

---

## Chapter 5

**RQ4** What are skills and how challenging is it to manually identify them in job postings?

**RQ5** To what extent does continuous pre-training on unlabeled job posting data improve skill extraction in English?

---

---

<sup>3</sup>See this specific example here: <http://data.europa.eu/esco/skill/75e48971-3242-4ea5-a1ba-4deec48ad41c>.

Resources such as ESCO are also multilingual, it contains 28 different European languages. Previous works on skill identification focus on the English language (Jia et al., 2018; Sayfullina et al., 2018; Tamburri et al., 2020), which could hinder local job seekers from finding an occupation suitable to their specific skills within their language community via online job platforms. In Chapter 6), we focus on the Danish labor market and annotate the Danish job postings in a similar approach using our own developed guidelines from Chapter 5.

Further categorization of skills can help us gain insights into talent demands. Skill classification is the task of further categorizing extracted skill spans. Existing label categorization could include ESCO, beyond these, there are alternative taxonomies, such as the Russian professional standard in Botov et al. (2019) or the Chinese Occupation Classification Grand Dictionary used in Cao and Zhang (2021). However, in this thesis, we make use of ESCO, due to its comprehensiveness and coverage of the English and Danish language. Instead of manually annotating the skill category for each span, one can leverage alternative approaches to induce labels automatically. This is generally known as distant supervision (Mintz et al., 2009), further defined in Subsection 2.4.3. We seek to answer:

---

## Chapter 6

**RQ6** How much does continuous pre-training on large-scale unlabeled Danish job posting data improve skill classification in Danish?

**RQ7** How effective is distant supervision with ESCO for skill categorization?

---

## 1.2 Modeling Occupational Skills (Part III)

It is practically infeasible to annotate all the job posting data for skills. Most SE approaches are supervised and need costly and time-consuming annotation. There are other ways to obtain labeled data, such as distant supervision, as mentioned in the previous section. We can leverage knowledge bases like ESCO to find matching skill spans in text. This can be approached using exact or fuzzy string matching, or more powerful methods that find similar skill spans in embedding space.

---

### Chapter 7

**RQ8** Can we apply existing skill labels from the ESCO taxonomy as a weak supervision signal for skill extraction?

---

Instead of using ESCO as an external knowledge base to match skills, there are several investigations on integrating factual knowledge or rare/infrequent words into a language model (Peters et al., 2019; Zhang et al., 2019; He et al., 2020; Wang et al., 2021b,a; Yu et al., 2022). We could instead integrate ESCO directly into a language model. As previous research has been focusing on English and Danish, we take it a step further and integrate all 28 languages of ESCO into a multilingual language model (e.g., Conneau et al., 2020) and improve performance for skill extraction and classification.

---

## Chapter 8

**RQ9** Can we use a graph-guided pre-training method with ESCO to improve a multilingual language model’s performance on skill extraction and classification?

---

Re-training or even further pre-training language models can be costly with the large compute necessary, especially in the context of large language models. One effective strategy to circumvent this is retrieval augmentation (Grave et al., 2017; Lee et al., 2019; Guu et al., 2020; Lewis et al., 2020b; Khandelwal et al., 2020; Izacard and Grave, 2021; Izacard et al., 2022). Here, language models can use external modules to enhance their context-processing ability. Two approaches are commonly used: First, using a separately trained model to retrieve relevant documents from a collection. This approach is employed in open-domain question answering tasks (Petroni et al., 2021).

Second, previous work on explicit memorization showed promising results with a cache (Grave et al., 2017), which serves as a type of datastore. The cache contains past hidden states of the model as keys and the next word as tokens in key-value pairs. Memorization of hidden states in a datastore, involves using the  $k$ NN algorithm as the retriever. The first work of the  $k$ -nearest neighbors ( $k$ NN) algorithm as the retrieval component was by Khandelwal et al. (2020), leading to several LM decoder-based works. Regardless, it is unclear how we can leverage multiple datasets for retrieval augmentation with encoder-based models.



---

## Chapter 9

**RQ10** How does a retrieval-augmented language model as a dataset-unifying method affect robustness for skill extraction?

---

### 1.3 Linking Skills to Existing Resources (Part IV)

Finally, once we extracted the skills, it is important to link surface form skills to a taxonomy entry, allowing us to quantify the current labor market dynamics and determine the demands and needs. This task can be framed as an entity linking problem (He et al., 2013; Logeswaran et al., 2019; Wu et al., 2020). Specifically, we aim at the linking of fine-grained span-level skill mentions to a specific taxonomy entry.

Generally, entity linking is the task of linking mentions of entities in unstructured text documents to their respective entities in a knowledge base, most commonly Wikipedia (He et al., 2013). Recent models address this problem by producing entity representations from a (sub)set of knowledge base information, e.g., entity descriptions (Logeswaran et al., 2019; Wu et al., 2020), fine-grained entity types (Raiman and Raiman, 2018; Onoe and Durrett, 2020; Ayoola et al., 2022), or generation of the input text autoregressively (Cao et al., 2021, 2022).

For linking skills to a taxonomy, we can use the European Skills, Competences, Qualifications, and Occupations (ESCO; le Vrang et al., 2014) taxonomy. Previous work classified spans to its taxonomy code via multi-class classification (Zhang et al., 2022b) without surrounding context and neither the full breadth of ESCO. Gnehm et al. (2022a) approaches this as a coarse-grained sequence labeling task using a subset of ESCO, and

others classify on the sentence level (Decorte et al., 2022, 2023; Clavié and Soulié, 2023). Therefore, we investigate:

---

## Chapter 10

**RQ11** How effective are entity linking methodologies for linking skills to knowledge bases (e.g., ESCO)?

---

## 1.4 Chapter Guide & Contributions

**Part II, III, and IV** contains the collection of peer-reviewed main contributions (detailed in [Section 1.5](#)), consisting of:

[Chapter 3](#), where we present **JOBSTACK**, a corpus for de-identification of personal data in job vacancies on Stackoverflow for ethical reasons and to comply with data regulations ([European Commission, 2016](#)). We introduce baselines, comparing BiLSTMs and Transformer models (**RQ1**). To improve upon these baselines, we experiment with LSTM models, Transformer models, and distantly related auxiliary data via multi-task learning. Our results show that auxiliary data from both generic named entity recognition data and medical data improves de-identification performance from job postings (**RQ2**).

In [Chapter 4](#), we introduce a novel Active Learning algorithm: Cartography Active Learning (CAL). It exploits the behavior of the model on individual instances *during training* as a proxy to find the most informative instances for labeling. CAL is inspired by data maps, which were proposed to derive insights into dataset quality ([Swayamdipta et al., 2020a](#)) (**RQ3**).

We introduce **SKILLSPAN** in [Chapter 5](#), a novel skill extraction dataset consisting of 14.5K sentences and over 12.5K annotated spans. We release its respective guidelines created over three different sources annotated for

hard *and* soft skills by domain experts (**RQ4**). Additionally, we continuously pre-train English-based language models on unlabeled job posting data and adapt them to the job market domain (**RQ5**).

In [Chapter 6](#), we release the first Danish job posting dataset: KOMPETENCER (*en*: competences), annotated for nested spans of competences. To improve upon coarse-grained annotations, we make use of the ESCO taxonomy API to obtain fine-grained labels via distant supervision. Additionally, we also further pre-train a Danish language model on unlabeled Danish job posting data and adapt it to the job market domain (**RQ6**, **RQ7**).

In [Chapter 7](#), we propose Skill Extraction with Weak Supervision. We leverage the ESCO taxonomy to find similar skills in job ads via latent representations. The method shows a strong positive signal, outperforming baselines based on token-level and syntactic patterns (**RQ8**).

There is a lack of generalized, multilingual models and benchmarks for skill extraction and classification tasks. In [Chapter 8](#), we introduce a language model called ESCOXML-R, based on XLM-R<sub>large</sub>, which uses domain-adaptive taxonomy-driven pre-training objective on the ESCO taxonomy, covering 27 languages. The pre-training objectives for ESCOXML-R include dynamic masked language modeling and a novel additional objective for inducing multilingual taxonomical ESCO relations (**RQ9**). We benchmark the model on nine job-related datasets in four different languages and shows that ESCOXML-R outperforms the existing state-of-the-art on six out of nine datasets.<sup>4</sup>

In [Chapter 9](#), we tackle the complexity in occupational skill datasets tasks—combining and leveraging multiple datasets for skill extraction—to identify rarely observed skills within a dataset, and overcoming the scarcity of skills across datasets. In particular, we investigate the retrieval-augmentation of language models, employing an external datastore for retrieving similar skills in a dataset-unifying manner. Our proposed method, **Nearest Neighbor Occupational Skill Extraction (NNOSE)** effectively leverages multiple datasets by retrieving neighboring skills from other datasets

---

<sup>4</sup>During the thesis, there were several contemporaneous datasets created and released.

in the datastore (**RQ10**). This improves skill extraction without additional fine-tuning.

Standardizing skill mentions can help us get insight into the current labor market demands by getting a distribution over the skill requirements in job ads. In **Chapter 10**, we explore Entity Linking (EL) in this domain, specifically targeting the linkage of occupational skills to the ESCO taxonomy (le Vrang et al., 2014) (**RQ11**). Previous efforts linked full sentences to a corresponding ESCO skill. Instead, in this work, we link fine-grained span-level mentions of skills.

## 1.5 Publications

The peer-reviewed work presented in this thesis consists of the following publications:

1. Kristian Nørgaard Jensen, Mike Zhang, and Barbara Plank. De-identification of privacy-related entities in job postings. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 210–221, Reykjavik, Iceland (Online), May 31–2 June 2021a. Linköping University Electronic Press, Sweden. URL <https://aclanthology.org/2021.nodalida-main.21>.<sup>5</sup> (**Chapter 3, RQ1, RQ2**)
2. Mike Zhang and Barbara Plank. Cartography active learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 395–406, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.36. URL <https://aclanthology.org/2021.findings-emnlp.36>. (**Chapter 4, RQ3**)
3. Mike Zhang, Kristian Jensen, Sif Sonniks, and Barbara Plank. SkillSpan: Hard and soft skill extraction from English job postings. In *Proceedings of the 2022 Conference of the North American Chapter of*

---

<sup>5</sup>Shared first authorship.

*the Association for Computational Linguistics: Human Language Technologies*, pages 4962–4984, Seattle, United States, July 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.366. URL <https://aclanthology.org/2022.naacl-main.366>. (Chapter 5, RQ4, RQ5)

4. Mike Zhang, Kristian Nørgaard Jensen, and Barbara Plank. Competencer: Fine-grained skill classification in Danish job postings via distant supervision and transfer learning. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 436–447, Marseille, France, June 2022b. European Language Resources Association. URL <https://aclanthology.org/2022.lrec-1.46>. (Chapter 6, RQ6, RQ7)
5. Mike Zhang, Kristian Nørgaard Jensen, Rob van der Goot, and Barbara Plank. Skill extraction from job postings using weak supervision. In *Proceedings of RecSysHR’22*. RecSysHR’22, 2022d. URL [https://ceur-ws.org/Vol-3218/RecSysHR2022-paper\\_10.pdf](https://ceur-ws.org/Vol-3218/RecSysHR2022-paper_10.pdf). (Chapter 7, RQ8)
6. Mike Zhang, Rob van der Goot, and Barbara Plank. ESCOXMLR: Multilingual Taxonomy-driven Pre-training for the Job Market Domain. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11871–11890, Toronto, Canada, 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.662. URL <https://aclanthology.org/2023.acl-long.662>. (Chapter 8, RQ9)
7. Mike Zhang, Rob van der Goot, Min-Yen Kan, and Barbara Plank. NNOSE: Nearest neighbor occupational skill extraction. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 589–608, St. Julian’s, Malta, March 2024a. Association for Computational

Linguistics. URL <https://aclanthology.org/2024.eacl-long.35>. (Chapter 9, RQ10)

8. Mike Zhang, Rob van der Goot, and Barbara Plank. Entity linking in the job market domain. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 410–419, St. Julian’s, Malta, March 2024b. Association for Computational Linguistics. URL <https://aclanthology.org/2024.findings-eacl.28>. (Chapter 10, RQ11)

All the papers above are covered in the thesis. I primarily led these articles in terms of framing the problem, conducting the experiments, and writing the papers with the help and supervision of my co-authors and supervisors.

### Other (Non-published) Articles

I was also part of several other works (some not published) that are not in this thesis.

9. Elisa Bassignana, Max Müller-Eberstein, Mike Zhang, and Barbara Plank. Evidence > intuition: Transferability estimation for encoder selection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4218–4227, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL <https://aclanthology.org/2022.emnlp-main.283>
10. Dennis Ulmer, Elisa Bassignana, Max Müller-Eberstein, Daniel Varab, Mike Zhang, Rob van der Goot, Christian Hardmeier, and Barbara Plank. Experimental standards for deep learning in natural language processing research. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2673–2692, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL <https://aclanthology.org/2022.findings-emnlp.196>
11. Maria Barrett, Max Müller-Eberstein, Elisa Bassignana, Amalie Brogaard Pauli, Mike Zhang, and Rob van der Goot. Can humans identify

domains?, 2024

12. Elena Senger, Mike Zhang, Rob van der Goot, and Barbara Plank. Deep learning-based computational job market analysis: A survey on skill extraction and classification from job postings. In *Proceedings of the First Workshop on Natural Language Processing for Human Resources (NLP4HR 2024)*, pages 1–15, St. Julian’s, Malta, March 2024. Association for Computational Linguistics. URL <https://aclanthology.org/2024.nlp4hr-1.1>
13. Antoine Magron, Anna Dai, Mike Zhang, Syrielle Montariol, and Antoine Bosselut. JobSkape: A framework for generating synthetic job postings to enhance skill matching. In *Proceedings of the First Workshop on Natural Language Processing for Human Resources (NLP4HR 2024)*, pages 43–58, St. Julian’s, Malta, March 2024. Association for Computational Linguistics. URL <https://aclanthology.org/2024.nlp4hr-1.4>
14. Khanh Nguyen, Mike Zhang, Syrielle Montariol, and Antoine Bosselut. Rethinking skill extraction in the job market domain using large language models. In *Proceedings of the First Workshop on Natural Language Processing for Human Resources (NLP4HR 2024)*, pages 27–42, St. Julian’s, Malta, March 2024. Association for Computational Linguistics. URL <https://aclanthology.org/2024.nlp4hr-1.3>





**Part I**

**Background**



# Chapter 2

## Challenges with Skill Extraction in the Job Market Domain

This chapter provides the general background for the thesis. We discuss how skills are defined, what the task of skill extraction entails, how we categorize skills, evaluate and analyze the predictions of skill extraction models, considerations for data annotation (e.g., privacy-related information and faster annotation), and how we can transfer general-purpose models to the job market domain.

### 2.1 Defining Skills

#### 2.1.1 The European Skills, Competences, Qualifications and Occupations

The notion of skill or competence encompasses diverse definitions within various disciplinary and research spheres (e.g., [Rauner et al., 2008](#); [Zhao and Rauner, 2014](#)). For example, [Rauner et al. \(2008\)](#) defines competences as possessing requisite abilities, authority, skills, and knowledge, reflecting an individual's psychological attributes. The concept of competence involves considerations of various influencing factors, including educational systems, cultural contexts, economic structures, vocational traditions, and

labor market dynamics.

Another definition of competences by [Peterßen \(2001\)](#) intends to capture specific reference points of human action. The differentiation follows the assumption that a person can be confronted with four types of challenges:

1. **Factual competence:** The focus is on dealing with material or symbolic objects, i.e., objects of nature or culture. Such objects include computers, office furniture, tools, or even texts, formulas, and programming languages.
2. **Social competence** is about dealing with other people in different communication situations (dyad, group/team, community).
3. **Self-competence** focuses on dealing with facets of one's own person. For example, it is about dealing with one's own emotions (e.g., fear, and aggression) or with one's own learning behavior.
4. **Methodological competence** is the usage of strategies and techniques of learned knowledge (i.e., factual competences).

As shown, competences are diverse, spurring substantial efforts toward categorization. The International Standard Classification of Occupations (ISCO; [Elias, 1997](#)) stands as a key global framework categorizing occupations and skills, operating within the international economic and social classification family. Similarly, the European Skills, Competences, Qualifications, and Occupations (ESCO; [le Vrang et al., 2014](#)) taxonomy, derived from ISCO, constitutes the European standard terminology connecting skills, competences, and qualifications to occupations. ESCO encompasses 13,890 competences and 3,008 occupations.<sup>1</sup> We show a graphical illustration of the hierarchy in ESCO in [Figure 2.1](#). Here, we only focused on a specific skill group (namely S4: management skills). The taxonomy is four levels deep, where skills in each level can be connected to an occupation.

---

<sup>1</sup>Per version 1.1.0.

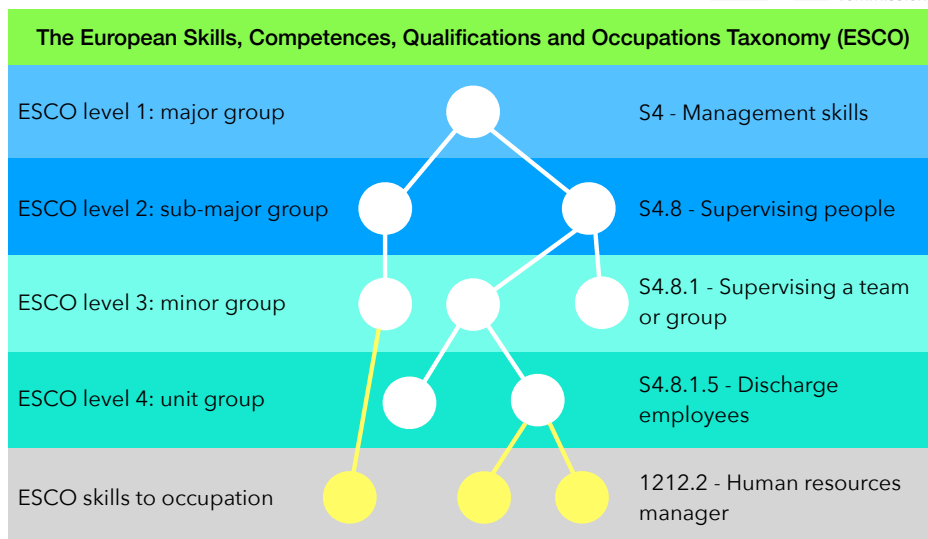


Figure 2.1: **A Graphical Illustration of ESCO.** We show the simplified structure of the hierarchical ESCO taxonomy. It contains four levels of skills, where each skill can be connected to a specific occupation.

At the time of the work done in this thesis divided competences into three categories: *Knowledge*, *Skill*, and *Attitudes*.

Knowledge, as per ESCO, is defined as the assimilation of information through learning, encompassing facts, principles, theories, and practices related to a particular field of work or study.<sup>2</sup> For instance, the acquisition of the Python programming language through learning represents a *knowledge* component, often referred to as a *hard skill*. Next, the application of this knowledge to specific tasks is regarded as a *skill* component, defined by ESCO as the ability to apply knowledge and utilize know-how to accomplish tasks and solve problems.<sup>3</sup> ESCO further designates *soft skills* as *attitudes*, considering them integral to skill components:

<sup>2</sup><https://ec.europa.eu/esco/portal/escopedia/Knowledge>

<sup>3</sup><https://ec.europa.eu/esco/portal/escopedia/Skill>

“The ability to use knowledge, skills, and personal, social and/or methodological abilities, in work or study situations and professional and personal development”<sup>4</sup>

In summary, *hard skills* are commonly identified as *knowledge* components, and the application of these skills is construed as a *skill* component. On the other hand, *soft skills*, termed as *attitudes* in ESCO, are integral to skill components.

A concrete example of a *knowledge* component is “K8: engineering, manufacturing and construction”, with one of the leaf nodes being “K8.3.1: architectural conservation”, this has the description “The practice of recreating forms, features, shapes, compositions, and architectural techniques of past constructions to preserve them”.<sup>5</sup>

A specific example of a *skill* component is “S6: handling and moving” with the description “Sorting, arranging, moving, transforming, fabricating and cleaning goods and materials by hand or using handheld tools and equipment. Tending plants, crops, and animals”. Then a sibling of this specific skill is “S6.2: moving and lifting” with the description “Performing physical activities to move, load, unload or store objects, position workpieces or equipment for assembly, or to climb structures, by hand or with the aid of equipment”.<sup>6</sup>

### 2.1.2 Challenges

The extraction of skills presents several distinctive nuances and challenges, which contribute to the complexity of this task and make it an intriguing area for exploration within the broader scope of NLP research.

1. There can be an implicitness in the representation of skills in textual

---

<sup>4</sup><https://data.europa.eu/esco/skill/A>

<sup>5</sup>See the specific knowledge component here <https://esco.ec.europa.eu/en/classification/skills?uri=http://data.europa.eu/esco/isced-f/0831>.

<sup>6</sup>See the specific skill example here <https://esco.ec.europa.eu/en/classification/skills?uri=http://data.europa.eu/esco/isced-f/08>.

content.<sup>7</sup> For instance, the skill of “teamwork” may manifest in various formats such as “being able to work together”, “working in a team”, or “collaborating efficiently”. The diverse expressions of these skills pose significant implications for the modeling approaches for SE.

2. Skills, by their nature, can exhibit considerable length, posing a challenge in terms of extraction with NLP models.
3. Furthermore, the distribution of skills follows an inherent long-tail pattern, signifying that certain skills are less frequently mentioned in a comprehensive set of job postings.
4. Another intriguing problem pertains to the discontinuity of skill spans, analogous to what is referred to as *coordinated conjunctions* in linguistics. This aspect introduces a layer of complexity in accurately delineating the boundaries of skills within the text.
5. Last, if we make use of a taxonomy like ESCO as the labels, this results in a substantial output space during the prediction phase, adding a layer of complexity to the modeling process.

We investigate and address several of these challenges in this thesis. First, in [Chapter 10](#) covers how well a model can link implicit skills to a taxonomical counterpart (Challenge 1). Second, in [Chapter 5](#) and [Chapter 7](#), we investigate how well encoder-based language models can predict lengthy skills (Challenge 2). Then, in [Chapter 9](#), we address how we can leverage multiple datasets to tackle the inherent long-tail frequency pattern of skills (Challenge 3). In [Chapter 6](#) and [Chapter 10](#), we look into the large label space of ESCO. We approach this challenge as both a classification and linking task (Challenge 5). Last, the fourth challenge of discontinuous skill extraction is not addressed in this thesis and is left for future work.

---

<sup>7</sup>In this context, an implicit skill refers to a standardized skill form that is not explicitly mentioned in the text.

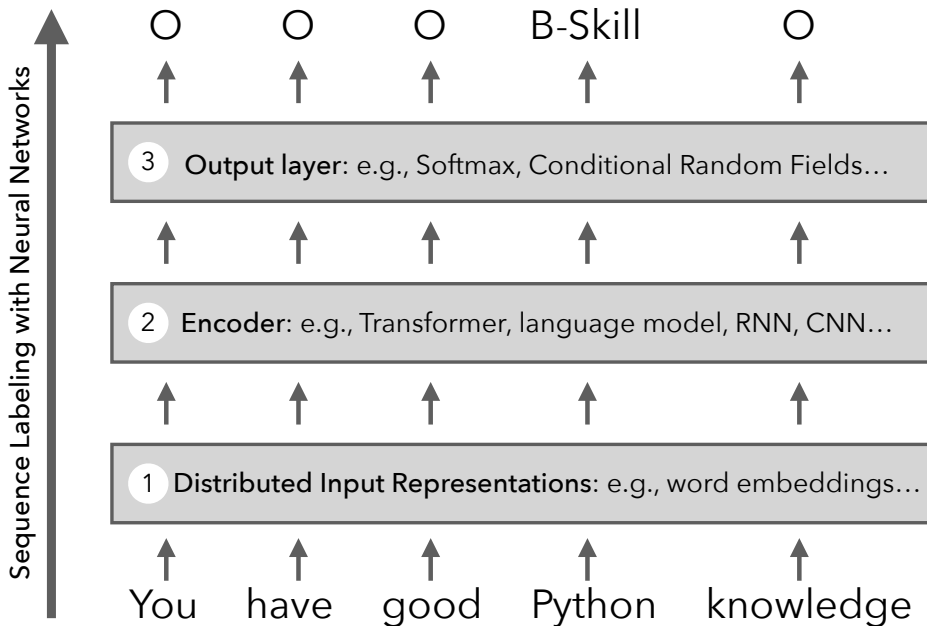


Figure 2.2: **Sequence Labeling with Neural Networks.** The flow for DL-based sequence labeling. The input sequence goes through 1) a layer that tokenizes the input, 2) an encoder that transforms the representations into meaningful vectors, 3) an output layer where the tags (B-Skill, I-Skill, O) get predicted from each token vector.



## 2.2 Skill Extraction

### 2.2.1 Skill Extraction Definition

In this thesis, the concept of skill extraction is operationalized by framing it as a sequence labeling problem as we want to extract the exact subspan from the text. Specifically, we use the BIO-labels (Ramshaw and Marcus, 1995) for token-level predictions. Formally, let  $\mathcal{D}$  represent a set of Job Descriptions (JDs), where each  $d \in \mathcal{D}$  constitutes a set of sequences, denoted as  $\mathcal{X}_d^i = \{x_1, x_2, \dots, x_T\}$  for the  $i^{\text{th}}$  input sequence, and a corresponding target sequence of BIO-labels,  $\mathcal{Y}_d^i = \{y_1, y_2, \dots, y_T\}$  (e.g., B-Skill”, I-Skill”, “O”). The primary objective is to train a sequence labeling algorithm, denoted as  $h : \mathcal{X} \mapsto \mathcal{Y}$ , utilizing  $\mathcal{D}$  to accurately predict entity spans by assigning an output label  $y_t$  to each token  $x_t$ .

**Other Definitions** Another frequently used task formulation of SE is multi-label classification (Bhola et al., 2020; Goyal et al., 2023). Formally, for a given input text  $t \in T$ , we seek a mapping  $f : T \rightarrow [0, 1]^S$ , where  $S = |S|$  represents the cardinality of the global skill set (i.e., predefined label space). The function  $f$  provides a probability score for each label  $s \in S$  given the input text  $t$ :

$$f(t) = P(r_i = 1|t)$$

where  $i \in \{1, \dots, S\}$ , and  $r_i$  denotes the label corresponding to the  $i$ -th skill.  $P$  is usually the result of a sigmoid function and ensures that the output of  $f(t)$  is constrained to the interval  $[0, 1]$ . To encapsulate the essence of the task, it entails assigning labels to textual descriptions from an extensive set of skills. However, in this thesis, we are mostly concerned with the sequence labeling task for skill extraction, as we can extract the exact subspan a skill label refers to from the text.

### 2.2.2 Span Extraction with Neural Networks

The task of sequence labeling is to assign a label to every token in a sentence. A single span could consist of multiple tokens within a sentence. The tokens in the sentence are usually represented in the BIO format (Beginning, Inside, and Outside) where every token is labeled as a B-label if the token is the beginning of the span, and I-label if it is the inside of the span, but not the first token (Ramshaw and Marcus, 1995; Tjong Kim Sang, 2002b).

In Figure 2.2, we show the common framework of a sequence labeling problem in context of neural networks. The input sentence “You have good Python knowledge” goes through three stages: First, the sentence gets transformed into some form of input representations, these are frequently lookup tables assigning vectors to corresponding tokens. Second, the input representations go through an encoder, where the goal is to learn a representation of the input sequence. Last, the encoded representations are transformed in the output layer, consisting usually of a feedforward layer and a softmax or Conditional Random Fields (CRF; Lafferty et al., 2001) layer to predict the final labels.

### 2.2.3 Evaluation

In this work, the evaluation of sequence labeling involves two primary evaluation schemes: *strict* and *loose* span-F1. We always use strict span-F1 (Tjong Kim Sang and De Meulder, 2003; Nakayama, 2018), same as standard span-F1, unless otherwise stated. The emphasis on span-F1, as opposed to regular F1, is driven by the correctness and annotation at the full span level. The evaluation criteria for strict and loose spans are different based on the matching boundaries between predicted and gold spans, i.e., true positives (TP).

**Strict** We count towards TP if the exact boundaries of predicted and gold spans match, including the correct skill type.

**Loose** We count a TP when there is an overlap between the boundaries of predicted and gold spans, including the correct skill type.

False positives (FP) are the predicted spans that are not in the gold evaluation set and false negatives (FN) are the gold spans that are not predicted. Then, to calculate precision and recall, we do the following:

**Precision** One can interpret this as “out of the predicted spans, how many are correct?”

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \text{ or} \quad (2.1)$$

$$\text{Precision} = \frac{\text{Number of correctly predicted spans}}{\text{Total number of predicted spans}},$$

**Recall** One can interpret this as “out of all gold spans, how many are correct?”

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \text{ or} \quad (2.2)$$

$$\text{Recall} = \frac{\text{Number of correctly predicted spans}}{\text{Total number of true spans}}$$

Then, the final Span-F1 score is calculated as the harmonic mean between precision and recall (Van Rijsbergen, 1979):

$$\text{Span-F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (2.3)$$

One could also consider other  $\beta$  weighting for the F-score, instead of a balanced weighting as in span-F1. For example, two common values for  $\beta$  are 2, which weighs recall higher than precision, and 0.5, which weighs recall lower than precision (Van Rijsbergen, 1979).

## 2.3 Categorization of Skills

Once skill spans are extracted from the text, we can further categorize them. In turn, this allows us to quantify skill demand more precisely (Section 1.3). For categorization, we use the ESCO taxonomy labels as the gold standard. We outline two types of skill categorization: 1) *Coarse-grained skill categorization* adopts the top-level major group of ESCO, yielding 23 labels (Subsection 2.3.1). 2) *Fine-grained skill categorization* involves the entire breadth of the taxonomy, encompassing around 14,000 skills, each with its unique taxonomy code (Subsection 2.3.2).

### 2.3.1 Coarse-grained Skill Categorization

Coarse-grained skill classification refers to the classification of already extracted skill spans from text, where we define coarse as 50 labels or less. Considering a set of job descriptions  $\mathcal{D}$ , where  $d \in \mathcal{D}$  is a collection of extracted spans (not full sentences), with the  $i^{\text{th}}$  span denoted as  $\mathcal{X}_d^i = \{x_1, x_2, \dots, x_T\}$  and a target class  $c \in \mathcal{C}$ , where  $\mathcal{C} = \{S^*, K^*\}$ .  $S^*$  and  $K^*$  are *skill* and *knowledge* components from the ESCO taxonomy (Subsection 2.1.1). The goal is to train an algorithm  $h : \mathcal{X} \mapsto \mathcal{C}$  to predict skill tags accurately by assigning an output label  $c$  for input  $\mathcal{X}_d^i$ .

**Evaluation of Skill Categorization** Given that some skills may be more prevalent than others (Aitor et al., 2003; Aitor and Dorn, 2013), an inherent class imbalance exists in the set of ground-truth skills. To take this imbalance into account, one can use the weighted macro-F1, a metric to evaluate a classification model by calculating the F1-score for each class individually, taking into account the number of instances in each class:

$$\text{Weighted Macro-F1} = \frac{\sum_{i=1}^C w_i \times \text{F1}_i}{\sum_{i=1}^C w_i} \quad (2.4)$$

where:

$C$  is the total number of classes,

$w_i$  is the weight, based on class frequency, assigned to class  $i$ ,

$F1_i$  is the F1 score for class  $i$ ,

$\text{Precision}_i$  is the precision for class  $i$ ,

$\text{Recall}_i$  is the recall for class  $i$ .

The F1 score for each class  $i$  is also calculated as the harmonic mean between precision and recall (Equation (2.3)).

### 2.3.2 Fine-grained Skill Categorization

In the context of skill categorization, this can be posed as an entity linking problem, where we attempt to match a span to an existing taxonomy code. Formally, we process the input document  $\mathcal{D} = \{w_1, \dots, w_r\}$ , a collection of entity mentions denoted as  $\mathcal{MD} = \{m_1, \dots, m_n\}$ , and a knowledge base (KB), ESCO in this case:  $\mathcal{E} = \{e_1, \dots, e_{13890}, \text{UNK}\}$ , where the number 13,890 is the total amount of skills in ESCO and UNK is a label for a non-linkable skill. The objective of an entity linking model is to generate a list of mention-entity pairs  $\{(m_i, e_i)\}_{i=1}^n$ , where each entity  $e$  corresponds to an entry in the KB.

**Evaluation of Skill Categorization** For the evaluation of linking performance, we assess the accuracy of generated mention-entity pairs in comparison to the ground truth. Here, we use the evaluation metric  $\text{Accuracy}@k$ , following prior research (Logeswaran et al., 2019; Wu et al., 2020; Zaprojets et al., 2022). We calculate the correctness between mentions and entities in the KB as the sum of correct hits or true positives if the ground truth for instance  $i$  is in the top- $k$  predictions, formally:

$$\text{Accuracy}@k = \frac{1}{n} \sum_{i=1}^n \text{TP in top-}k \text{ for instance } i. \quad (2.5)$$

In summary, in this section, we formalized the task of skill extraction and closely related tasks. In the next sections, we will be talking about several peculiarities in SE and how we analyze them.

## 2.4 Annotating Data

This section will provide a concise overview of practical considerations within data annotation. Specifically, we will delve into the considerations of handling personally identifiable information (PII) within annotated datasets, recognizing its significance in ethical and privacy dimensions. Additionally, we will explore the incorporation of active learning techniques to enhance efficiency in data labeling.

### 2.4.1 Personally Identifiable Information

Job advertisement data contains information that pertains to individual privacy. This data typically comprises predominantly *general* information about individuals, as opposed to *sensitive* information, as defined by the European General Data Protection Regulation (GDPR; [European Commission, 2016](#)). General information encompasses details such as names, addresses, email addresses, phone numbers, and similar identifiers. On the other hand, sensitive information includes data related to racial or ethnic backgrounds, religious beliefs, social security numbers, and so forth. Ensuring the privacy of individuals is a legal mandate, enforced by various legislations such as the US Health Insurance Portability and Accountability Act (HIPAA; [Act, 1996](#)) and the GDPR. For this thesis, our concern is primarily with complying with ethical and legal requirements to safeguard the privacy of subjects, specifically about the general information contained in job advertisements. In [Chapter 3](#), we investigate whether we can train an NLP system to remove parts of the text that can lead to the identification of individuals.

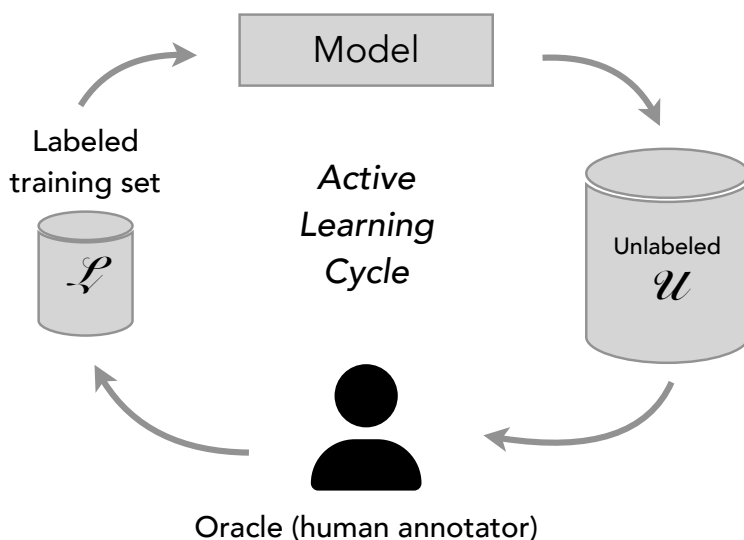


Figure 2.3: **Pool-based Active Learning Cycle.** The human annotator starts to annotate a small labeled set  $\mathcal{L}$  to train an initial model. Then, the model is applied to an unlabeled pool of data  $\mathcal{U}$ . Given a certain acquisition function or score, the model selects the instances that are the most *informative*. This set of instances is given to the oracle to annotate.

### 2.4.2 Active Learning

The main idea of Active Learning (AL) is that a machine learning model can achieve greater accuracy with fewer labeled training instances if it can *choose* the data from which it learns. Generally, it is a widely-used method to tackle the time-consuming and expensive collection and manual labeling of data (Cohn et al., 1994; Lewis and Catlett, 1994; Settles, 2009). One can consider injecting such a process into an annotation cycle. An example of an AL cycle (pool-based in this case) is shown in Figure 2.3. We start with a human annotator (also called an oracle) to annotate a couple of examples. The number of initial examples varies, but usually, the initial labeled set  $\mathcal{L}$  should be enough to give a machine learning some signal of the task. Once the model is trained, this is applied to an unlabeled data pool  $\mathcal{U}$ . Given some acquisition function or score, the model selects the instances having the highest *informativeness*. This set of unlabeled

examples is given to the oracle for annotation. This cycle continues for a couple of rounds until a desired performance is achieved or other stopping criteria (e.g., convergence). In [Chapter 4](#), we introduce a novel acquisition function, where we show a new form of estimating *informativeness* during model training.

### 2.4.3 Weak Supervision

In machine learning, a significant challenge comes from the demand for extensive sets of manually annotated data. Weak supervision emerges as a strategic solution to address this issue by leveraging human expertise to generate noisy labels, often termed as *weak* labels ([Stephan and Roth, 2022](#); [Zhu et al., 2023a](#)). This approach commonly involves the incorporation of keywords, heuristics, or insights from external data sources, usually called *distant supervision* ([Craven et al., 1999](#); [Mintz et al., 2009](#)).

In this thesis, we interchangeably use the terms weak supervision and distant supervision. However, these can be interpreted as one being a part of another, where distant supervision is a type of weak supervision ([Stephan and Roth, 2022](#)). In [Chapter 6](#) and [Chapter 7](#), we leverage the ESCO Application Programming Interface (API) in two different approaches: First, in [Chapter 6](#), we directly query the API with free form text to obtain noisy taxonomy codes for skill spans. Then, in [Chapter 7](#), we extract the taxonomy entries from ESCO, and try to match them to n-grams in job posting text via heuristics (e.g., exact or fuzzy string matching) and embedding approaches.

## 2.5 Analysis

There are a couple peculiarities in skill extraction that can be linked back to the challenges mentioned in [Subsection 2.1.2](#), such as the long length of skills and the long-tail distribution of skills. In this section, we discuss several approaches to analysis for skill extraction that have been used in several chapters of this thesis.



### 2.5.1 Span Length

Skills may be of considerable length, as will be demonstrated in [Chapter 5](#), where manually annotated skills were observed to span up to 20 tokens.<sup>8</sup> Our focus lies in assessing the predictive efficacy of language models, specifically in the context of handling long skills. To gauge the robustness of these language models on span length, we adopt a structured approach by categorizing the gold standard spans into different buckets based on their token length. Subsequently, we calculate the Span-F1 ([Equation \(2.3\)](#)) score for each distinct bucket.

Formally, let  $\{B_1, B_2, \dots, B_k\}$  represent the token length buckets, where each  $B_i$  corresponds to a range of token lengths. The gold standard spans are then assigned to the appropriate bucket  $B_i$  based on their token length. The Span-F1 score is computed independently for each bucket, providing an evaluation of the language model’s predictive performance across varying lengths of skills. This approach allows us to gain insights into model performance concerning the length of the predicted spans, thereby improving our understanding of its capabilities in handling diverse skill lengths. We use this type of analysis in [Chapter 5](#) and [Chapter 8](#).

### 2.5.2 Long-tail Skills

The distribution of skills across different industries can vary, indicating a tendency for certain skills to be more popular than others. To ensure that our models account for the natural patterns of skill occurrence, we investigate how models perform on skills with lower frequency ([Chapter 8](#), [Chapter 9](#)). Similar to our approach with span length, we categorize gold standard spans based on their frequency counts and proceed to calculate the Span-F1 ([Equation \(2.3\)](#)) score for each frequency bucket.

Formally, let  $S_g$  be the set of gold standard spans in the dataset, and

---

<sup>8</sup>One example here could be: “assist local authors in writing and delivering articles while ensuring that local articles are in high quality and that language is at eye level with our employees and comply with corporate standards”. Note that this skill is also part of the *coordinated conjunction* challenge mentioned in [Subsection 2.1.2](#).

$f(s_i)$  represent the frequency count of each skill  $s_i$ . We define non-overlapping frequency buckets  $B = \{B_1, B_2, \dots, B_k\}$  based on the distribution of gold standard spans. A function  $\phi : S_g \rightarrow B$  assigns each span  $s_i$  to the corresponding frequency bucket  $B_j$  based on  $f(s_i)$ . This approach allows us to gain insights into whether the model performs well even under limited data of skills. We use this analysis in [Chapter 8](#) and [Chapter 9](#).

### 2.5.3 False Positives

The ESCO taxonomy, comprised of 13,890 labels, defines a finite label space. There is, however, a “field unknown” label for knowledge components. Continuous improvement of ESCO is crucial to sustain its value for the labor market, as well as education and training systems. Employers consistently demand new emerging occupations and skills, while changes in curricula and terminology are regularly integrated into education and training programs. To effectively address these, it is imperative to actively share feedback, suggestions, and proposals on enhancing the content and management of the classification. Organizations using ESCO, along with other stakeholders, play a key role in this collaborative effort to ensure the ongoing relevance and effectiveness of ESCO. When confronted with the emergence of skills within the ESCO taxonomy, one way of finding them involves inspecting instances of false positives generated by the model. False positives, in this context, denote predictions that deviate from the gold annotations, typically indicative of overprediction tendencies. However, a qualitative inspection, as undertaken in various works in this thesis ([Chapter 8](#), [Chapter 9](#), and [Chapter 10](#)), reveals instances where the model generates predictions resembling skills. This qualitative analysis indicates that a manual inspection of false positives can be useful for finding new skills or missed annotations and getting realistic estimations of performance.

## 2.6 Transfer to the Job Market Domain

### 2.6.1 Motivation

Most contemporary NLP systems are designed and trained with a focus on generic data (e.g., Wikipedia or large-scale web data). A fundamental problem in machine learning is that (self-)supervised learning systems strongly depend on the data they were trained on. Changes in the characteristics of this training data has strong effects on model transferability (Kittredge and Grisham, 1986). This issue of “domain divergence” has prompted comprehensive surveys on how to best adapt language models trained on one domain to more specific targets (Ramponi and Plank, 2020; Ramesh Kashyap et al., 2021; Saunders, 2022), and remains an open issue, even with language models of increasing size (Ling et al., 2023; Singhal et al., 2023; Wu et al., 2023). This is no different from text in the job market domain. In Chapter 5, Chapter 6, Chapter 8, and Chapter 9, we show that models that are tuned on domain-specific data (i.e., job descriptions) outperform models that have been pre-trained on general data.

### 2.6.2 Multi-task Learning

Machine learning generally involves training a model to perform a single task. However, a model can benefit by learning from auxiliary, related tasks. Multitask Learning (MTL; Caruana, 1997) is an inductive transfer mechanism. The goal is to improve generalization performance. MTL improves generalization by leveraging the domain or task-specific information contained in the training signals of related tasks. It does this by training tasks in parallel while sharing (a portion of) its weights. In effect, the training signals for the extra tasks serve as an inductive bias. The seminal work by Caruana (1997) mostly discusses the concept of *hard parameter sharing*: In neural network-based models, it is usually applied by sharing the hidden layers (i.e., the input representations and encoder in Figure 2.2) between all tasks while keeping task-specific output layers (Ruder, 2017), which we use in this thesis. In Chapter 3, we show such an example of MTL for

the task of de-identification in job postings. We combine several datasets from different domains (e.g., medical) and investigate whether multi-task learning is beneficial and improves on the target task of de-identification.

### 2.6.3 Transfer Learning

In the conventional supervised machine learning framework, the premise is to train a model for a specific task and domain. This entails having labeled data corresponding to the task and domain in question. By using this labeled dataset, we can train a model with the expectation that it will exhibit proficient performance when applied to previously unseen data associated with the same task and domain. Essentially, our expectation is rooted in the assumption that the data is independently and identically distributed. Conversely, when presented with data for an alternative task or domain, we once again need labeled data specific to that task or domain. This new set of labeled data is then used to train a distinct model, with the expectation that it will excel when applied to data of this particular type.

Once there is an insufficient amount of labeled data available for the intended task or domain, this will hinder the training of a reliable model. Transfer learning emerges as a solution to address this limitation, allowing us to tackle such scenarios by capitalizing on data from a related task or domain, termed the source task and source domain. Instead of training the model to perform multiple tasks simultaneously to improve the performance of another related task as in multi-task learning, transfer learning aims to transfer knowledge learned from one task (source task) to improve performance on a different, but related, task (target task). In practical terms, the objective is to transfer as much knowledge as possible from the source context to our target task or domain. The nature of this knowledge can manifest in diverse forms, contingent upon the specific task and data at hand. For example, we have the same task (e.g., skill extraction) in one language (e.g., English) and we investigate the performance of the English-based model in another language, for example, Danish. This is an example of *transductive transfer learning* and specifically *cross-lingual transfer learn-*

*ing*, where the source and target downstream tasks are the same (Blitzer et al., 2007; Arnold et al., 2007), but the source and target domains are different (Ruder, 2017), i.e., the text is written in a different language. We explore cross-lingual transfer learning in Chapter 6 and Chapter 8.



**Part II**

**Annotating Data**





# Chapter 3

## De-identification of Privacy Related Entities in Job Postings

The work presented in this chapter is based on a paper that has been published as: Kristian Nørgaard Jensen, Mike Zhang, and Barbara Plank. De-identification of privacy-related entities in job postings. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 210–221, Reykjavik, Iceland (Online), May 31–2 June 2021a. Linköping University Electronic Press, Sweden. URL <https://aclanthology.org/2021.nodalida-main.21>.

## Abstract

De-identification is the task of detecting privacy-related entities in text, such as person names, emails and contact data. It has been well-studied within the medical domain. The need for de-identification technology is increasing, as privacy-preserving data handling is in high demand in many domains. In this paper, we focus on job postings. We present JOBSTACK, a new corpus for de-identification of personal data in job vacancies on Stackoverflow. We introduce baselines, comparing Long-Short Term Memory (LSTM) and Transformer models. To improve upon these baselines, we experiment with contextualized embeddings and distantly related auxiliary data via multi-task learning. Our results show that auxiliary data improves de-identification performance. Surprisingly, vanilla BERT turned out to be more effective than a BERT model trained on other portions of Stackoverflow.

## 3.1 Introduction

It is becoming increasingly important to anonymize privacy-related information in text, such as person names and contact details. The task of de-identification is concerned with detecting and anonymizing such information. Traditionally, this problem has been studied in the medical domain by e.g., [Szarvas et al. \(2007\)](#); [Friedrich et al. \(2019\)](#); [Trienes et al. \(2020\)](#) to anonymize (or pseudo-anonymize) person-identifiable information in electronic health records (EHR). With new privacy regulations ([Section 3.2](#)) de-identification is becoming more important for broader types of text. For example, a company or public institution might seek to de-identify documents before sharing them. On another line, de-identification can benefit society and technology at scale. Particularly auto-regressive models trained on massive text collections pose a potential risk for exposing private or sensitive information ([Carlini et al., 2019, 2020](#)), and de-identification can be one way to address this.

In this paper, we analyze how effective sequence labeling models are in

identifying privacy-related entities in job posts. To the best of our knowledge, we are the first study that investigates de-identification methods applied to job vacancies. In particular, we examine:

- How do Transformer-based models compare to LSTM-based models on this task?
- How does BERT compare to BERT<sub>Overflow</sub> (Tabassum et al., 2020)
- To what extent can we use existing medical de-identification data and Named Entity Recognition (NER) data to improve de-identification performance?

To answer these questions, we put forth a new corpus, JOBSTACK, annotated with around 22,000 sentences in English job postings from Stackoverflow for person names, contact details, locations, and information about the profession of the job post itself.

**Contributions.** We present JOBSTACK, the first job postings dataset with professional and personal entity annotations from Stackoverflow. Our experiments on entity de-identification with neural methods show that Transformers outperform bi-LSTMs, but surprisingly a BERT variant trained on another portion of Stackoverflow is less effective. We find auxiliary tasks from both news and the medical domain to help boost performance.

## 3.2 Related Work

### 3.2.1 De-identification in the Medical Domain

De-identification has mostly been investigated in the medical domain (e.g., Szarvas et al. (2007); Meystre et al. (2010); Liu et al. (2015); Jiang et al. (2017); Friedrich et al. (2019); Trienes et al. (2020)) to ensure the privacy of a patient in the analysis of their medical health records. Apart from an ethical standpoint, it is also a legal requirement imposed by multiple legislations such as the US Health Insurance Portability and

Accountability Act (HIPAA; [Act, 1996](#)) and the European General Data Protection Regulation (GDPR; [European Commission, 2016](#)).

Many prior works in the medical domain used the I2B2/UTHealth dataset ([Stubbs and Uzuner, 2015b](#)) to evaluate de-identification. The dataset consists of clinical narratives, which are free-form medical texts written as a first person account by a clinician. Each of the documents describes a certain event, consultation or hospitalization. All of the texts have been annotated with a set of Protected Health Information (PHI) tags (e.g. name, profession, location, age, date, contact, IDs) and subsequently replaced by realistic surrogates. The dataset was originally developed for use in a shared task for automated de-identification systems. Systems tend to perform very well on this set, in the shared task three out of ten systems achieved F1 scores above 90 ([Stubbs and Uzuner, 2015b](#)). More recently, systems reach over 98 F1 with neural models ([Dernoncourt et al., 2017](#); [Liu et al., 2017b](#); [Khin et al., 2018](#); [Trienes et al., 2020](#); [Johnson et al., 2020](#)). We took I2B2 as inspiration for annotation of JOBSTACK.

Past methods for de-identification in the medical domain can be categorised in three categories. (1) Rule-based approaches, (2) traditional machine learning (ML)-based systems (e.g., feature-based Conditional Random Fields (CRFs; [Lafferty et al., 2001](#)), ensemble combining CRF and rules, data augmentation, clustering), and (3) neural-based approaches.

### 3.2.2 Rule-based

First, [Gupta et al. \(2004\)](#) made use of a set of rules, dictionaries, and fuzzy string matching to identify protected health information (PHI). In a similar fashion, [Neamatullah et al. \(2008\)](#) used lexical look-up tables, regular expressions, and heuristics to find instances of PHI.

### 3.2.3 Traditional ML

Second, classical ML approaches employ feature-based CRFs ([Aberdeen et al., 2010](#); [He et al., 2015](#)). Moreover, earlier work showed the use of CRFs in an ensemble with rules ([Stubbs and Uzuner, 2015b](#)). Other ML

approaches include data augmentation by [McMurry et al. \(2013\)](#), where they added public medical texts to properly distinguish common medical words and phrases from PHI and trained decision trees on the augmented data.

### 3.2.4 Neural methods

Third, regarding neural methods, [Dernoncourt et al. \(2017\)](#) were the first to use Bi-LSTMs, which they used in combination with character-level embeddings. Similarly, [Khin et al. \(2018\)](#) performed de-identification by using a Bi-LSTM-CRF architecture with ELMo embeddings ([Peters et al., 2018](#)). [Liu et al. \(2017b\)](#) used four individual methods (CRF-based, Bi-LSTM, Bi-LSTM with features, and rule-based methods) for de-identification, and used an ensemble learning method to combine all PHI instances predicted by the three methods. [Trienes et al. \(2020\)](#) opted for a Bi-LSTM-CRF as well, but applied it with contextual string embeddings ([Akbik et al., 2018](#)). Most recently, [Johnson et al. \(2020\)](#) fine-tuned  $BERT_{\text{base}}$  and  $BERT_{\text{large}}$  ([Devlin et al., 2019b](#)) for de-identification. Next to “vanilla” BERT, they experiment with fine-tuning different domain specific pre-trained language models, such as SciBERT ([Beltagy et al., 2019](#)) and BioBERT ([Lee et al., 2020a](#)). They achieve state-of-the-art performance in de-identification on the I2B2 dataset with the fine-tuned  $BERT_{\text{large}}$  model. From a different perspective, the approach of [Friedrich et al. \(2019\)](#) is based on adversarial learning, which automatically pseudo-anonymizes EHRs.

### 3.2.5 De-identification in other Domains

Data protection in general however is not only limited to the medical domain. Even though work outside the clinical domain is rare, personal and sensitive data is in abundance in all kinds of data. For example, [Eder et al. \(2019\)](#) pseudonymized German emails. [Bevendorff et al. \(2020\)](#) published a large preprocessed email corpus, where only the email addresses themselves were anonymized. Apart from emails, several works went into the de-identification of SMS messages ([Treurniet et al., 2012](#); [Patel et al., 2013](#);

```

...
13. Job description:
14. [XXXOrganization] is a modern multi tenant, microservices based solution and
Floor Planning is one major functional solution vertical of the [XXXOrganization]
platform.
15. What you'll be doing:
16. As a [XXXProfession] for [XXXOrganization], you will be one of the founding
members of our [XXXLocation] based floor planning development team.
17. You will be in charge for development of future floor planning capabilities on
the [XXXOrganization] platform and be the software architect for the capability.
18. You will drive the team to improve the coding practices and boost performance.
19. You will also be a member of our [XXXOrganization] and have a major influence
on feature roadmap and technologies we use.
...

```

Figure 3.1: **Snippet JOBSTACK**. Snippet of a job posting, full job posting can be found in [Section 3.7](#).

[Chen and Kan, 2013](#)) in Dutch, French, English, and Mandarin respectively. Both [Treurniet et al. \(2012\)](#); [Chen and Kan \(2013\)](#) conducted the same strategy and automatically anonymized all occurrences of dates, times, decimal amounts, and numbers with more than one digit (telephone numbers, bank accounts, et cetera), email addresses, URLs, and IP addresses. All sensitive information was replaced with a placeholder. [Patel et al. \(2013\)](#) introduced a system to anonymize SMS messages by using dictionaries. It uses a dictionary of first names and anti-dictionaries (of ordinary language and of some forms of SMS writing) to identify the words that require anonymization. In our work, we study de-identification for names, contact information, addresses, and professions, as further described in [Section 3.3](#).

### 3.3 JOBSTACK

In this section, we describe the JOBSTACK dataset. There are two basic approaches to removing privacy-bearing data from job postings. First, anonymization identifies instances of personal data (e.g. names, email addresses, phone numbers) and replaces these strings with some placeholder (e.g. {name}, {email}, {phone}). The second approach, pseudonymization

preserves the information of personal data by replacing these privacy-bearing strings with randomly chosen alternative strings from the same privacy type (e.g. replacing a name with “John Doe”). The term de-identification subsumes both anonymization and pseudonymization. In this work, we focus on anonymization.<sup>1</sup>

Eder et al. (2019) argues that the anonymization approach might be appropriate to eliminate privacy-bearing data in the medical domain, but would be inappropriate for most Natural Language Processing (NLP) applications since crucial discriminative information and contextual clues will be erased by anonymization.

If we shift towards pseudonymization, we argue that there is still the possibility to resurface the original personal data. Henceforth, our goal is to anonymize job postings to the extent that one would not be able to easily identify a company from the job posting. However, as job postings are public, we are aware that it would be simple to find the original company that posted it with a search engine. Nevertheless, we abide to GDPR compliance which requires us to protect the personal data and privacy of EU citizens for transactions that occur within EU member states (European Commission, 2016). In job postings, this would be the names of employees, and their corresponding contact information.<sup>2</sup>

Over a period of time, we scraped 2,755 job postings from Stackoverflow and selected 395 documents to annotate, the subset ranges from June 2020 to September 2020. We manually annotated the job postings with the following five entities: Organization, Location, Contact, Name, and Profession.

To make the task as realistic as possible, we kept all sentences in the documents. The statistics provided in the following therefore reflect the natural distribution of entities in the data. A snippet of an example job post can be seen in Figure 3.1, the full job posting can be found in

---

<sup>1</sup>Meystre (2015) notes that de-identification means removing or replacing personal identifiers to make it difficult to reestablish a link between the individual and his or her data, but it does not make this link impossible.

<sup>2</sup><https://t.ly/yVjEq>

Timespan Docs.→	06–08.2020		09.2020		Total
Statistic↓	Train	Dev	Test		
# Documents	313	41	41		395
# Sentences	18,055	2082	2092		22,219
# Tokens	195,425	22,049	21,579		239,053
# Entities	4,057	462	426		5,154
average # sentences	57.68	50.78	51.02		53.16
average tokens / sent.	10.82	10.59	10.32		10.78
average entities / sent.	0.22	0.22	0.20		0.21
density	14.73	14.31	14.58		14.54
Organization	1803	215	208		2226
Location	1511	157	142		1810
Profession	558	63	64		685
Contact	99	10	7		116
Name	86	17	5		108

Table 3.1: **Statistics of JOBSTACK.** We document the statistics of our JOBSTACK dataset. We show the surface-level statistics (e.g., the number of documents). We also show several sentence-level statistics (e.g., average tokens per sentence). Last, we note the number of the entities annotated in our dataset.

### Subsection 3.7.1.

#### 3.3.1 Statistics

Table 3.1 shows the statistics of our dataset. We split our data in 80% train, 10% development, and 10% test. Besides of a regular document-level random split, ours is further motivated based on time. The training set covers the job posts posted between June to August 2020 and the development- and test set are posted in September 2020. To split the text into sentences, we use the `sentence-splitter` library used for processing the Europarl corpus (Koehn, 2005). In the training set, we see that the average number of sentences is higher than in the development- and test set (6-7 more). We therefore also calculate the density of the entities, meaning the percentage of sentences with at least one entity. The table shows that



14.5% of the sentences in JOBSTACK contain at least one entity. Note that albeit having document boundaries, we treat the task of de-identification as a standard word-level sequence labeling task.

### 3.3.2 Annotation Schema

The aforementioned entity tags are based on the English I2B2/UTHealth corpus (Stubbs and Uzuner, 2015b). The tags are more coarse-grained than the I2B2 tags. For example, we do not distinguish between zip code and city, but tag them with `Location`. We give a brief explanation of the tags.

**Organization:** This includes all companies and their legal entity mentioned in the job postings. The tag is not limited to the company that authored the job posting but does also include mentions of stakeholders or any other company.

**Location:** This is the address of the company in the job posting. The location also refers to all other addresses, zip codes, cities, regions, and countries mentioned throughout the text. This is not limited to the company address, but should be used for all location names in the job posting, including abbreviations.

**Contact:** The label includes, URLs, email addresses and phone numbers. This could be, but is not limited to, contact info of an employee from the authoring company.

**Name:** This label covers the names of people. This could be, but is not limited to, a person from the company, such as the contact person, CEO, or manager. All names appearing in the job posting should be annotated no matter the relation to the job posting itself. Titles such as Dr. are not part of the annotation. Apart from people names in our domain, difficulties could arise with other types of names. An example would be project names, with which one could identify a company. In this work, we did not annotate such names.

**Profession:** This label covers the profession that is being searched for in the job posting or desired prior relevant jobs for the current profession. We

	Token	Entity	Unlabeled
A1 – A2	0.889	0.767	0.892
A1 – A3	0.898	0.782	0.904
A2 – A3	0.917	0.823	0.920
Fleiss’ $\kappa$	0.902	0.800	0.906

Table 3.2: **Inter-annotator Agreement.** We show agreement over pairs with Cohen’s  $\kappa$  and all annotators with Fleiss’  $\kappa$ .

do not annotate additional meta information such as gender (e.g. Software Engineer (f/m)). We also do not annotate mentions of colleague positions in either singular or plural form. For example: “As a *Software Engineer*, you are going to work with *Security Engineers*”. Here we annotate Software Engineer as profession, but we do not annotate Security Engineers. While this may sound straightforward, however, there are difficulties in regards to annotating professions. A job posting is free text, meaning that one can write anything they prefer to make the job posting as clear as possible (e.g., *Software Engineer (at a unicorn start-up based in [..])*). The opposite is also possible when they are looking for one applicant to fill in one of multiple positions. For example, “We are looking for an applicant to fill in the position of *DevOps/Software Engineer*”. From our interpretation, they either want a “DevOps Engineer” or a “Software Engineer”. We decided to annotate the full string of characters “DevOps/Software Engineer” as a profession.

### 3.3.3 Annotation Quality

To evaluate our annotation guidelines, a sample of the data was annotated by three annotators, one with a background in Linguistics (A1) and two with a background in Computer Science (A2, A3). We used an open-source text annotation tool named Doccano (Nakayama et al., 2018). There are around 1,500 overlapping sentences that we calculated agreement on. The annotations were compared using Cohen’s  $\kappa$  (Fleiss and Cohen, 1973) between pairs of annotators, and Fleiss’  $\kappa$  (Fleiss, 1971), which generalizes Cohen’s  $\kappa$  to more than two concurrent annotations. Table 3.2 shows

three levels of  $\kappa$  calculations, we follow [Balasuriya et al. \(2009\)](#)'s approach of calculating agreement in NER. (1) `Token` is calculated on the token level, comparing the agreement of annotators on each token (including non-entities) in the annotated dataset. (2) `Entity` is calculated on the agreement between named entities alone, excluding agreement in cases where all annotators agreed that a token was not a named entity. (3) `Unlabeled` refers to the agreement between annotators on the exact span match over the surface string, regardless of the type of named entity (i.e., we only check the position of tag without regarding the type of the named entity). [Landis and Koch \(1977\)](#) state that a  $\kappa$  value greater than 0.81 indicates almost perfect agreement. Given this, all annotators are in strong agreement.

After this annotation quality estimation, we finalized the guidelines. They formed the basis for the professional linguist annotator, who annotated and finalized the entire final `JOBSTACK` dataset.

## 3.4 Methods

For entity de-identification we use a classic Named Entity Recognition (NER) approach using a Bi-LSTM with a CRF layer. On top of this we evaluate the performance of Transformer-based models with two different pre-trained BERT variants. Furthermore, we evaluate the helpfulness of auxiliary tasks, both using data close to our domain, such as de-identification of medical notes, and more general NER, which covers only a subset of the entities. Further details on the data are given in [Subsection 3.4.3](#).

### 3.4.1 Models

Firstly, we test a Bi-LSTM sequence tagger (Bilty) ([Plank et al., 2016](#)), both with and without a CRF layer. The architecture is similar to the widely used models in previous works. For example, preliminary results of Bilty versus [Trienes et al. \(2020\)](#) show accuracy almost identical to each other: 99.62% versus 99.76%. Next, we test a Transformer based

model, namely the MaChAmp (van der Goot et al., 2021c) toolkit. Current research shows good results for NER using a Transformer model without a CRF layer (Martin et al., 2020a), hence we tested MaChAmp both with and without a CRF layer for predictions. For both models, we use their default parameters.

### 3.4.2 Embeddings

For embeddings, we tested with no pre-trained embeddings, pre-trained Glove (Pennington et al., 2014) embeddings, and Transformer-based pre-trained embeddings. For Transformer-based embeddings, we focused our attention on two BERT models, BERT<sub>base</sub> (Devlin et al., 2019b) and BERT<sub>Overflow</sub> (Tabassum et al., 2020). When using the Transformer-based embeddings with the Bi-LSTM, the embeddings were fixed and did not get updated during training.

Using the MaChAmp (van der Goot et al., 2021c) toolkit, we fine-tune the BERT variant with a Transformer encoder. For the Bi-LSTM sequence tagger, we first derive BERT representations as input to the tagger. The tagger further uses word and character embeddings which are updated during model training.

The BERT<sub>Overflow</sub> model is a transformer with the same architecture as BERT<sub>base</sub>. It has been trained from scratch on a large corpus of text from the Q&A section of Stackoverflow, making it closer to our text domain than the “vanilla” BERT model. However, BERT<sub>Overflow</sub> is not trained on the job postings portion of Stackoverflow.

### 3.4.3 Auxiliary tasks

Both the Bi-LSTM (Plank et al., 2016) and the MaChAmp (van der Goot et al., 2021c) toolkit are capable of multi-task Learning (MTL; Caruana, 1997). We, therefore, set up a number of experiments testing the impact of three different auxiliary tasks. The auxiliary tasks and their datasets are as follows:

Model	Precision	Recall	F1-score
Bilty	79.00 $\pm$ 1.10	65.80 $\pm$ 3.72	71.76 $\pm$ 2.57
Bilty + CRF	84.09 $\pm$ 1.90	67.96 $\pm$ 0.81	75.15 $\pm$ 0.66
Bilty + Glove 50d	79.21 $\pm$ 2.19	67.03 $\pm$ 2.76	72.53 $\pm$ 0.83
Bilty + Glove 50d + CRF	82.93 $\pm$ 0.87	64.93 $\pm$ 3.93	72.74 $\pm$ 2.23
Bilty + BERT <sub>base</sub>	83.70 $\pm$ 0.58	73.01 $\pm$ 1.34	77.99 $\pm$ 0.91
Bilty + BERT <sub>base</sub> + CRF	<b>88.23</b> $\pm$ 0.87	73.30 $\pm$ 1.47	80.09 $\pm$ 0.60
Bilty + BERT <sub>Overflow</sub>	70.86 $\pm$ 0.68	41.27 $\pm$ 4.19	52.01 $\pm$ 3.15
Bilty + BERT <sub>Overflow</sub> + CRF	77.79 $\pm$ 1.20	40.33 $\pm$ 2.98	53.08 $\pm$ 2.88
MaChAmp + BERT <sub>base</sub>	86.66 $\pm$ 0.73	84.78 $\pm$ 0.44	85.70 $\pm$ 0.13
MaChAmp + BERT <sub>base</sub> + CRF	86.40 $\pm$ 0.62	<b>86.15</b> $\pm$ 0.00	<b>86.27</b> $\pm$ 0.31
MaChAmp + BERT <sub>Overflow</sub>	70.88 $\pm$ 0.17	61.47 $\pm$ 0.81	65.84 $\pm$ 0.48
MaChAmp + BERT <sub>Overflow</sub> + CRF	77.27 $\pm$ 3.68	63.06 $\pm$ 2.11	69.35 $\pm$ 0.96

Table 3.3: **JOBSTACK Development Set Results.** Results on the development set across three runs using our JOBSTACK dataset.

- I2B2/UTHealth (Stubbs and Uzuner, 2015b) - Medical de-identification;
- CoNLL 2003 (Tjong Kim Sang, 2002a) - News Named Entity Recognition;
- The combination of the above.

The data of the two tasks are similar to our dataset in two different ways. The I2B2 lies in a different text domain, namely medical notes, however, the label set of the task is close to our label set, as mentioned in Subsection 3.3.2. For CoNLL, we have a general corpus of named entities but fewer types (location, organization, person, and miscellaneous), but the text domain is presumably closer to our data. We test the impact of using both auxiliary tasks along with our own dataset.

Model	Auxiliary tasks	Precision	Recall	F1 Score
Bilty + BERT <sub>base</sub> + CRF	JOBSTACK + CoNLL	86.91 $\pm$ 1.94	77.49 $\pm$ 1.87	81.90 $\pm$ 0.32
	JOBSTACK + I2B2	83.61 $\pm$ 2.61	75.18 $\pm$ 2.59	79.15 $\pm$ 2.19
	JOBSTACK + CoNLL + I2B2	84.92 $\pm$ 1.67	78.28 $\pm$ 4.34	81.37 $\pm$ 2.01
Bilty + BERT <sub>Overflow</sub> + CRF	JOBSTACK + CoNLL	79.34 $\pm$ 2.34	46.54 $\pm$ 1.99	58.62 $\pm$ 1.46
	JOBSTACK + I2B2	72.03 $\pm$ 6.48	46.10 $\pm$ 2.55	55.99 $\pm$ 1.93
	JOBSTACK + CoNLL + I2B2	71.20 $\pm$ 4.80	50.86 $\pm$ 3.31	59.15 $\pm$ 2.15
MaChAmp + BERT <sub>base</sub> + CRF	JOBSTACK + CoNLL	87.24 $\pm$ 1.94	<b>87.23<math>\pm</math>1.24</b>	<b>87.20<math>\pm</math>0.34</b>
	JOBSTACK + I2B2	<b>88.44<math>\pm</math>0.84</b>	84.92 $\pm$ 0.44	86.64 $\pm$ 0.53
	JOBSTACK + CoNLL + I2B2	86.13 $\pm$ 0.50	86.00 $\pm$ 0.87	86.06 $\pm$ 0.66
MaChAmp + BERT <sub>Overflow</sub> + CRF	JOBSTACK + CoNLL	75.65 $\pm$ 1.41	66.24 $\pm$ 0.98	70.62 $\pm$ 0.64
	JOBSTACK + I2B2	80.26 $\pm$ 1.32	68.47 $\pm$ 1.03	73.88 $\pm$ 0.16
	JOBSTACK + CoNLL + I2B2	77.66 $\pm$ 0.82	69.41 $\pm$ 0.89	73.29 $\pm$ 0.22

Table 3.4: **Multi-task Learning Results.** Performance of multi-task learning on the development set across three runs.

Model	Auxiliary tasks	Precision	Recall	F1 Score
Bilty + BERT <sub>base</sub> + CRF	JOBSTACK	<b>82.44<math>\pm</math>0.95</b>	75.90 $\pm$ 1.39	78.99 $\pm$ 0.32
MaChAmp + BERT <sub>base</sub> + CRF	JOBSTACK	75.92 $\pm$ 0.39	84.35 $\pm$ 0.49	79.91 $\pm$ 0.38
	JOBSTACK + CoNLL	77.84 $\pm$ 1.19	85.06 $\pm$ 0.91	81.27 $\pm$ 0.28
	JOBSTACK + I2B2	80.30 $\pm$ 0.99	83.88 $\pm$ 0.67	<b>82.05<math>\pm</math>0.80</b>
	JOBSTACK + CoNLL + I2B2	77.66 $\pm$ 0.58	<b>85.68<math>\pm</math>0.57</b>	81.47 $\pm$ 0.43

Table 3.5: **Test Set Results.** Evaluation of the best-performing models on the test set across three runs.

## 3.5 Evaluation

All results are mean scores across three different runs.<sup>3</sup> The metrics are all calculated using the `conlleval` script<sup>4</sup> from the original CoNLL-2000 shared task. Table 3.3 shows the results from training on JOBSTACK only, Table 3.4 shows the results of the MTL experiments described in Subsection 3.4.3. Both report results on the development set. Lastly, Table 3.5 shows the scores from evaluating selected best models as found on the development set, when tested on the final held-out test set.

<sup>3</sup>We sampled three random seeds: 3477689, 4213916, 8749520 which are used for all experiments.

<sup>4</sup><https://www.clips.uantwerpen.be/conll2000/chunking/output.html>

### 3.5.1 Is a CRF layer necessary?

In [Table 3.3](#), as expected, adding the CRF for the Bi-LSTM clearly helps, and consistently improves precision and thereby F1 score. For the stronger BERT model the overall improvement is smaller and does not necessarily stem from higher precision. We note that on average across the three seed runs, MaChAmp with BERT<sub>base</sub> and no CRF mistakenly adds an I-tag following an O-tag 8 times out of 426 gold entities. In contrast, the MaChAmp with BERT<sub>base</sub> and CRF, makes no such mistake in any of its three seed runs. Earlier research, such as [Souza et al. \(2019\)](#) show that BERT models with a CRF layer improve or perform similarly to its simpler variants when comparing the overall F1 scores. Similarly, they note that in most cases it shows higher precision scores but lower recall, as in our results for the development set. However, interestingly, the precision drops during test for the Transformer-based model. As the overall F1 score increases slightly, we use the CRF layer in all subsequent experiments. The main takeaway here is that both models benefit from an added CRF layer for the task, but the Transformer model to a smaller degree.

### 3.5.2 LSTM versus Transformer

Initially, LSTM networks dominated the de-identification field in the medical domain. Up until recently, large-scale pre-trained language models have been ubiquitous in NLP, although rarely used in this field. On both development and test results ([Table 3.3](#), [Table 3.5](#)), we show that a Transformer-based model outperforms the LSTM-based approaches with non-contextualized and contextualized representations.

### 3.5.3 Poor performance with BERT<sub>Overflow</sub>

BERT<sub>base</sub> is the best embedding method among all experiments using Bilty, with BERT<sub>Overflow</sub> being the worst with a considerable margin. Being able to fine-tune BERT<sub>base</sub> does give a good increase in performance overall. The same trend is apparent with fine-tuning BERT<sub>Overflow</sub>, but it is not

enough to catch up with  $BERT_{base}$ . We see that overall MaChAmp with  $BERT_{base}$  and CRF is the best model. However, Bilty with  $BERT_{base}$  and CRF does have the best precision.

We hypothesized the domain-specific  $BERT_{Overflow}$  representations would be beneficial for this task. Intuitively,  $BERT_{Overflow}$  would help with detecting profession entities. Profession entities contain specific skills related to the IT domain, such as *Python developer*, *Rust developer*, *Scrum master*. Although the corpus it is trained on is not one-to-one to our vacancy domain, we expected to see at most a slight performance drop. This is not the case, as the drop in performance turned out to be high. It is not fully clear to us why this is the case. It could be the Q&A data it is trained on consists of more informal dialogue than in job postings. In the future, we would like to compare these results to train a BERT model on job postings data.

### 3.5.4 Auxiliary data increases performance

Looking at the results from the auxiliary experiments in [Table 3.4](#) we see that all auxiliary sources are beneficial, for both types of models. A closer look reveals that once again MaChAmp with  $BERT_{base}$  is the best performer across all three auxiliary tasks. Also, we see that Bilty with  $BERT_{base}$  has good precision, though not the best this time around. For a task like de-identification recall is preferable, thereby showing that fine-tuning BERT is better than the classic Bi-LSTM-CRF. Moreover, we see that  $BERT_{Overflow}$  is under-performing compared to  $BERT_{base}$ . However,  $BERT_{Overflow}$  is able to get a 4 point increase in F1 with I2B2 as auxiliary task in MaChAmp. For Bilty with  $BERT_{Overflow}$  we see a slightly greater gain with both CoNLL and I2B2 as auxiliary tasks. When comparing the auxiliary data sources to each other, we note that the closer text domain (CoNLL news) is more beneficial than the closer label set (I2B2) from a more distant medical text source. This is consistent for the strongest models.

In general, it can be challenging to train multi-task networks that outperform or even match their single-task counterparts ([Martínez Alonso](#)



and Plank, 2017; Clark et al., 2019). Ruder (2017) mentions training on a large number of tasks is known to help regularize multi-task models. A related benefit of MTL is the transfer of learned “knowledge” between closely related tasks. In our case, it has been beneficial to add auxiliary tasks to improve our performance on both development and test compared to a single-task setting. In particular, it seemed to have helped with pertaining a high recall score.

### 3.5.5 Performance on the test set

Finally, we evaluate the best performing models on our held out test set. The best models are selected based on their performance on F1, precision, and recall. The results are seen in Table 3.5. Comparing the results to those seen in Table 3.3 and Table 3.4 it is clear to see that Bilty with BERT<sub>base</sub> sees a smaller drop in F1 compared to that of MaChAmp with BERT<sub>base</sub>. We do also see an increase in recall for Bilty compared to its performance on the development set. In general, we see that recall for each model is staying quite stable without any significant drops. It is also interesting to see that, the internal ranking between MTL MaChAmp with BERT<sub>base</sub> has changed, with JOBSTACK + I2B2 being the best-performing model in terms of F1.

### 3.5.6 Per-entity Analysis

In Table 3.6, we show a deeper analysis on the test set: the performance of the two different auxiliary tasks in a multi-task learning setting, namely CoNLL and I2B2. We hypothesized different performance gains with each auxiliary task. For I2B2, we expected Contact and Profession to do better than CoNLL, since I2B2 contains contact information entities (e.g., phone numbers, emails, et cetera) and professions of patients. Surprisingly, this is not the case for Contact, as CoNLL outperforms I2B2 on all three metrics. We do note however this result could be due to little instances of Contact and Name being present in the gold test set. Additionally, both named entities are predicted six to nine times by both models on all three runs

Entity Type↓	MaChAmp + JOBSTACK		
		+ CoNLL	+ I2B2
<b>Organization</b> (208)	F1	77.51 ± 0.81	78.34 ± 1.32
	P	73.73 ± 1.66	77.86 ± 1.60
	R	81.73 ± 0.96	78.85 ± 1.74
<b>Location</b> (142)	F1	86.88 ± 1.51	86.67 ± 1.80
	P	83.86 ± 1.82	83.47 ± 1.19
	R	90.14 ± 1.41	90.14 ± 2.54
<b>Profession</b> (64)	F1	80.20 ± 2.76	83.88 ± 0.90
	P	77.44 ± 3.82	82.42 ± 0.63
	R	83.33 ± 4.51	85.42 ± 1.80
<b>Contact</b> (7)	F1	87.91 ± 3.81	75.48 ± 4.30
	P	90.47 ± 8.25	71.03 ± 4.18
	R	85.71 ± 0.00	80.95 ± 8.24
<b>Name</b> (5)	F1	86.25 ± 8.08	85.86 ± 4.38
	P	76.39 ± 12.03	75.40 ± 6.87
	R	100.00 ± 0.00	100.00 ± 0.00

Table 3.6: Performance of the two different auxiliary tasks. Reported is the F1, Precision (P), and Recall (R) per entity. The number behind the entity name is the gold label instances in the test set.

on the test set. This could indicate a strong difference in performance. For Profession, it shows that I2B2 is beneficial for this particular named entity as expected. For the other three named entities, the performance is similar. As Location, Name, and Organization are in both datasets, we did not expect any difference in performance. The results confirm this intuition.

## 3.6 Conclusions

In this work, we introduce JOBSTACK, a dataset for de-identification of English Stackoverflow job postings. Our implementation is publicly avail-

able.<sup>5</sup> The dataset is freely available upon request. We present neural baselines based on LSTM and Transformer models. Our experiments show the following: (1) Transformer-based models consistently outperform Bi-LSTM-CRF-based models that have been standard for de-identification in the medical domain. (2) Stackoverflow-related BERT representations are not more effective than regular BERT representations on Stackoverflow job postings for de-identification. (3) MTL experiments with BERT representations and related auxiliary data sources improve our de-identification results; the auxiliary task trained on the closer text type was the most beneficial, yet results improved with both auxiliary data sources. This shows the benefit of using multi-task learning for de-identification in job vacancy data.

---

<sup>5</sup><https://github.com/kris927b/JobStack>

## 3.7 Appendix

### 3.7.1 Example Job Posting

Below, we show an example snippet of a de-identified job posting, there is a cut-off for brevity.

1. [XXX<sub>Profession</sub>]
2. [XXX<sub>Organization</sub>]
3. <ADDRESS>, <ADDRESS>, [XXX<sub>Location</sub>] , - , [XXX<sub>Location</sub>]  
<cutoff>
  
13. Job description:
14. [XXX<sub>Organization</sub>] is a modern multi tenant, microservices based solution and Floor Planning is one major functional solution vertical of the [XXX<sub>Organization</sub>] platform.
  
15. What you'll be doing:
16. As a [XXX<sub>Profession</sub>] for [XXX<sub>Organization</sub>], you will be one of the founding members of our [XXX<sub>Location</sub>] based floor planning development team.
17. You will be in charge for development of future floor planning capabilities on the [XXX<sub>Organization</sub>] platform and be the software architect for the capability.
18. You will drive the team to improve the coding practices and boost performance.
19. You will also be a member of our [XXX<sub>Organization</sub>] and have a major influence on feature roadmap and technologies we use.
  
20. What you'll bring to the table:  
<cutoff>
28. Ability to work on-site in our [XXX<sub>Location</sub>] office, with flexible remote work possibilities  
<cutoff>

# Chapter 4

## Cartography Active Learning

The work presented in this chapter is based on a paper that has been published as: Mike Zhang and Barbara Plank. Cartography active learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 395–406, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.36. URL <https://aclanthology.org/2021.findings-emnlp.36>.

## Abstract

We propose Cartography Active Learning (CAL), a novel Active Learning (AL) algorithm that exploits the behavior of the model on individual instances *during training* as a proxy to find the most informative instances for labeling. CAL is inspired by data maps, which were recently proposed to derive insights into dataset quality (Swayamdipta et al., 2020a). We compare our method on popular text classification tasks to commonly used AL strategies, which instead rely on post-training behavior. We demonstrate that CAL is competitive to other common AL methods, showing that training dynamics derived from small seed data can be successfully used for AL. We provide insights into our new AL method by analyzing batch-level statistics utilizing the data maps. Our results further show that CAL results in a more data-efficient learning strategy, achieving comparable or better results with considerably less training data.

## 4.1 Introduction

Active Learning (AL) is a widely-used method to tackle the time-consuming and expensive collection and manual labeling of data. In recent years, many AL strategies were proposed. The simplest and most widely used is *uncertainty sampling* (Lewis and Gale, 1994; Lewis and Catlett, 1994), where the learner queries instances that it is most uncertain about. Uncertainty sampling is myopic: it only measures the information content of a single data instance. Alternative AL algorithms instead focus on selecting a *diverse* batch (Geifman and El-Yaniv, 2017; Sener and Savarese, 2018; Gissin and Shalev-Shwartz, 2019; Zhdanov, 2019) or to estimate the *uncertainty* distribution of the learner (Houlsby et al., 2011; Gal and Ghahramani, 2016). However, these methods are usually limited in their notion of informativeness, which is tied to post-training model uncertainty and batch diversity.

Recently, Swayamdipta et al. (2020a) introduced *data maps*, to visualize the behaviour of the model on individual instances during training

(*training dynamics*). The plotted data maps (Figure 4.1) reveal distinct regions in a dataset: groups of *ambiguous* instances useful for high performance and linked to high informativeness, *easy-to-learn* instances which aid optimization, and *hard-to-learn* instances which frequently correspond to mislabeled or erroneous instances.

We propose *Cartography Active Learning* (CAL), which automatically selects the the most informative instances that contribute optimally to model learning. To do so, we leverage a largely ignored source of information: insights derived *during training*, i.e., training dynamics derived from *limited* data maps (see Section 4.4) to choose informative instances at the boundary of ambiguous and hard-to-learn instances. We hypothesize that this region is where the model will *learn the most* from. Data maps provide the additional benefit that we can use them to measure informativeness of a batch with straightforward metrics and visualize dataset properties. These distinct regions in the data maps have their own respective statistics. Therefore, as a second research question we investigate whether data map statistics help to assess why some AL algorithms work better than others.

**Contributions.** In this paper, our contributions are twofold. (1) We present Cartography Active Learning, a novel AL algorithm that exploits data maps for AL. We compare our results against other competitive and widely used AL algorithms and outperform them in early AL iterations. (2) Additionally, we leverage the data maps to inspect what instances AL methods select. We show that our approach optimally selects informative instances avoiding only *hard-to-learn* and *easy-to-learn* cases, which leads to better AL and comparable or better results than full dataset training.

## 4.2 Related Work

AL has seen many usage scenarios in the Natural Language Processing (NLP) field (Shen et al., 2018; Lowell et al., 2019; Ein-Dor et al., 2020; Margatina et al., 2021). The perspective of AL is that if a model is allowed to select the data from which it will *learn the most*, it will achieve compa-

able (or better) performance with less training instances (Siddhant and Lipton, 2018), and at the same time addressing the costly labeling process with a human annotator.

A popular scenario is pool-based active learning (Lewis and Gale, 1994; Settles, 2009, 2012), which assumes a small set of labeled data  $\mathcal{L}$  and a large pool of unlabeled data  $\mathcal{U}$ . Most AL algorithms start similarly: a model is fit to  $\mathcal{L}$  to get access to  $P_\theta(y | x)$ , then apply a query strategy to get the best scored instance from  $\mathcal{U}$ , label this instance and add it to  $\mathcal{L}$  in an iterative process.

### 4.2.1 Common Strategies

A commonly used query strategy is *uncertainty sampling* (Lewis and Gale, 1994; Lewis and Catlett, 1994). In this approach, the learner queries the instances which it is least certain about. There are two popular approaches. (1) Uncertainty sampling based on entropy (Shannon, 1948; Dagan and Engelson, 1995), it uses the entropy of the label distribution as a measure for the uncertainty of the model on an instance. (2) Uncertainty sampling based on which best labeling is the least confident (Culotta and McCallum, 2005).

### 4.2.2 Batch-mode Active Learning

It is inefficient and time-consuming to obtain sampled queries one by one for annotation in the context of Deep Neural Networks (DNNs). In a real-world setting, consider having multiple annotators available. One can exploit this setting and label the instances in batches and parallel. Batch-mode AL allows the learner to query instances in groups. To assemble the optimal batch, one can greedily pick the top- $k$  examples according to an instance-level acquisition function suitable for DNNs. There are many works on ways for making neural network posteriors accurately represent the confidence on a given example. One popular example is stochastic regularisation techniques such as dropout during inference time, known as the Monte Carlo Dropout technique (Houlsby et al., 2011). Gal and



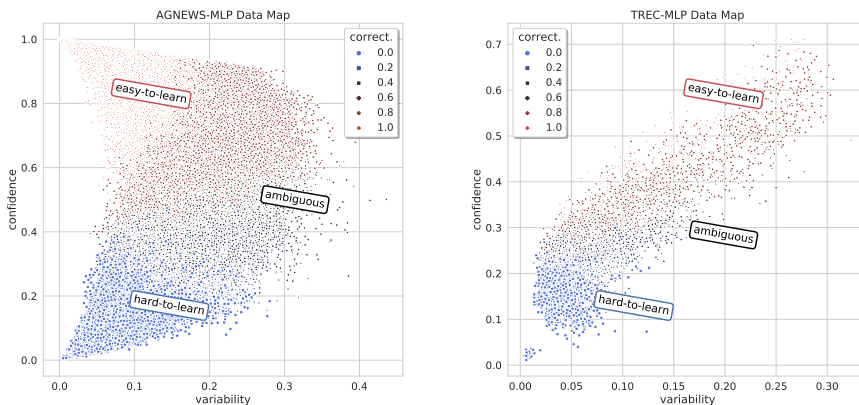


Figure 4.1: **Full Data Maps for AGNews & TREC.** AGNews (120,000 instances) on the left, and TREC (5,452 instances) on the right, both w.r.t. an MLP training for ten epochs. The x-axis shows **variability** and the y-axis the **confidence**. The colors and shapes indicate the **correctness**.

Ghahramani (2016) refer to this as Bayesian Active Learning by Disagreement (BALD). This allows us to consider the model as a Bayesian neural network and calculate approximations of uncertainty estimates by analyzing its multiple predictions. However, if the information of these top- $k$  examples is similar, this will result in the model not generalizing well over the dataset. Therefore, alternative approaches take the *diversity* of a batch into account.

### 4.2.3 Batch-aware Query Strategies

Instead of greedily choosing the examples that maximize some score, one can instead try to find a batch that is as diverse as possible. One recently proposed effective strategy is Discriminative Active Learning (DAL; Gissin and Shalev-Shwartz, 2019). This approach aims to select instances from  $\mathcal{U}$  that make  $\mathcal{L}$  representative of  $\mathcal{U}$ . In other words, the idea is to train a separate model to classify between  $\mathcal{L}$  and  $\mathcal{U}$ . Then, to use that model to choose the instances which are most confidently classified as being from  $\mathcal{U}$ . If  $\mathcal{U}$  and  $\mathcal{L}$  become indistinguishable, the learner has successfully

closed the data gap between  $\mathcal{U}$  and  $\mathcal{L}$ . DAL was proposed for computer vision and was recently successfully used in NLP (Ein-Dor et al., 2020). Alternative diversity AL strategies exist, such as core-set, which often rely on heuristics (Sener and Savarese, 2018; Geifman and El-Yaniv, 2017).<sup>1</sup>

### 4.3 Cartography Active Learning

The key idea of CAL is to use model-independent measures, from fitting the model on the seed data  $\mathcal{L}$ , by using data maps (Swayamdipta et al., 2020a) for AL. Data maps help identify characteristics of instances within the broader trends of a dataset by leveraging their training dynamics (i.e., the behavior of a model *during training*, such as mean and standard deviation of confidence and correctness with respect to the gold label). These model-dependent measures reveal distinct regions in a data map, by and large, reflecting instance properties (see Figure 4.1 and details below on *easy-to-learn*, *ambiguous*, and *hard-to-learn* instances). Training dynamics encapsulate information of data quality that has been largely ignored in AL: the sweet spot of instances at the boundary of *hard-to-learn* and *ambiguous* instances, which are quick to label while providing informative samples, as shown in full data training (Swayamdipta et al., 2020a).

In the next part, we introduce training characteristics, first showing the resulting data maps on the full data. Then we introduce CAL, which proposes to learn a data map from the seed labeled data  $\mathcal{L}$  and identifying regions of instances with a binary classifier, inspired by DAL (Gissin and Shalev-Shwartz, 2019). To identify these regions, we require: (1) a data map can be learned from limited data, and (2) a classifier to identify informative instances. The full algorithm, illustrated in Algorithm 1, is described later.

---

<sup>1</sup>Contemporary to our work and both appearing at EMNLP 2021, Margatina et al. (2021) proposed CAL (Contrastive Active Learning), another AL method with the same acronym to our approach. The main difference to Cartography AL is using data points that are similar in the model feature space, while optimizing for maximally disagreeing predictive likelihoods.

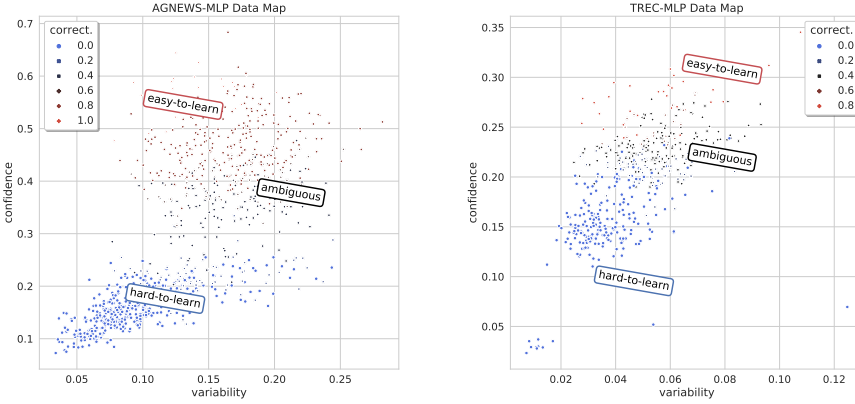


Figure 4.2: **Data Maps with Limited Seed Data.** Data map for the AGNews seed set (1,000 instances), and TREC seed set (500 instances). Both data maps are based on an MLP trained for ten epochs.

### 4.3.1 Mapping the Data

Formally, the training dynamics of instance  $i$  are defined as the statistics calculated over  $E$  epochs. These statistics are then used as the coordinates in the plot. The following statistics are calculated, **confidence**, **variability**, and **correctness**, following the notation of [Swayamdipta et al. \(2020a\)](#):

$$\hat{\mu}_i = \frac{1}{E} \sum_{e=1}^E p_{\theta^{(e)}}(y_i^* | \mathbf{x}_i) \quad (4.1)$$

**Confidence**<sup>2</sup> (Equation 4.1) is the mean model probability of the gold label ( $y_i^*$ ) across epochs. Where  $p_{\theta^{(e)}}$  is the model’s probability with parameters  $\theta^{(e)}$  at the end of the  $e^{\text{th}}$  epoch.

$$\hat{\sigma}_i = \sqrt{\frac{\sum_{e=1}^E (p_{\theta^{(e)}}(y_i^* | \mathbf{x}_i) - \hat{\mu}_i)^2}{E}} \quad (4.2)$$

<sup>2</sup>Similar to [Swayamdipta et al. \(2020a\)](#), we note that the term *confidence* here is the output probability of the model over the gold label as opposed to the certainty of the predicted label as commonly used in AL literature.

Then, **variability** (Equation 4.2) is calculated as the standard deviation of  $p_{\theta^{(e)}}(y_i^* | \mathbf{x}_i)$ , the spread across epochs  $E$ .

$$\hat{\phi}_i = \frac{1}{E} \sum_{e=1}^E \mathbf{1}(\hat{y}_i = y_i^* | \mathbf{x}_i) \quad (4.3)$$

Last, **correctness** (Equation 4.3) is denoted as the fraction of times the model correctly labels instance  $\mathbf{x}_i$  across epochs  $E$ .

Given the aforementioned training dynamics and the obtained statistics per instance, we plot the data maps for both AGNews (Zhang et al., 2015) and TREC (Li and Roth, 2002), using all training data (Figure 4.1). The data map is based on a Multi-layer Perceptron (MLP). As shown by Swayamdipta et al. (2020a), data maps identify three distinct regions: *easy-to-learn*, *ambiguous*, and *hard-to-learn*. The *easy-to-learn* instances are consistently predicted correctly with high confidence, these instances can be found in the upper region of the plot. The *ambiguous* samples have high variability and the model is inconsistent in predicting these correctly (middle region). The instances that are (almost) never predicted correctly, and have low confidence and variability, are referred to as *hard-to-learn* cases. This confirms findings by Swayamdipta et al. (2020a) where they show that training on the samples of these distinct regions, and in particular the *ambiguous* instances, promote optimal performance. While uncertainty-based AL mostly focus on hard-cases, CAL instead focuses on ambiguous and possibly easier instances.

### 4.3.2 Data Maps from Seed Data

Given the distinct regions for data selection in the full data map in Figure 4.1, we first investigate whether regions are still identifiable if we have little amounts of training data, as this is a prerequisite for CAL. Figure 4.2 shows this for 1,000 training samples of AGNews and 500 of TREC. We can see that the data points are more scattered, where the *easy-to-learn* and *ambiguous* samples are mixed. However, it seems the *hard-to-learn* region can still qualitatively be distinguished from the other regions.

### 4.3.3 CAL Algorithm

The algorithm is detailed in [Algorithm 1](#) and described next. To select presumably informative instances, we train a binary classifier on the seed set  $\mathcal{L}$  and apply it to  $\mathcal{U}$  to select the instances that are the closest to the decision boundary between *ambiguous* and *hard-to-learn* instances. By visualizing a decision function in [Figure 4.2](#) that separates the *hard-to-learn* region from the *ambiguous/easy-to-learn* region, we select the instances that are the closest to this boundary. In other words, selecting instances with output probability 0.5 with respect to the binary classifier. This does two things, (1) it prevents the binary classifier from selecting only *easy-to-learn* instances (low-variability, high-confidence), and (2) selecting some truly *hard-to-learn* instances (low-variability, low-confidence) which are both not optimal for learning.

Similar to DAL, for the binary classification task, we map our original input space  $\mathcal{X}$  to the learned representation  $\Psi$  of the last hidden layer of an MLP ([Subsection 4.4.3](#)). These are the features used in our binary classifier ( $\theta'$ ). Formally, as we have three hidden layers,

$$\Psi : \mathcal{X} \rightarrow \hat{\mathcal{X}}, \text{ where } \Psi = \mathbf{h}_3 = f(W_3 \cdot \mathbf{h}_2 + \mathbf{b}_3).$$

For label space  $\mathcal{Y}$ , we consider the binary values  $\{0, 1\}$ . This label depends on the *correctness*. The label  $y_{\Psi(\hat{\mathbf{x}}_i)}$  for the learned representation of instance  $\hat{\mathbf{x}}_i$  is labeled 1 when the *correctness* using the limited data map at epoch  $E$  is above the threshold  $t_{cor} > 0.2$ . We refer to these as **high-cor** cases. The samples that are rarely correct ( $t_{cor} \leq 0.2$ ) are labeled as 0 and we refer to these as **low-cor** cases. To give a better intuition, we refer to [Figure 4.2](#), where the regions of *hard-to-learn* and *ambiguous/easy-to-learn* are visually separable with this *correctness* threshold  $t_{cor} = 0.2$ . This threshold is empirically chosen by investigating the influence of different *correctness* thresholds on the performance of CAL in [Subsection 4.5.3](#) ([Table 4.3](#)).

---

**Algorithm 1:** Cartography Active Learning

---

```

1 input: Labeled seed set  $\mathcal{L}$ , Unlabeled set  $\mathcal{U}$ , Total budget  $K$ ,
   Number of queries  $n$ , Correctness threshold  $t_{cor} = 0.2$ ;
2 for  $i = 1, \dots, n$  do
3    $\Psi(\mathcal{L}), \Psi(\mathcal{U}) \leftarrow$  train main classifier  $\theta$  on  $\mathcal{L}$ , get representations
   of  $\mathcal{L}$  and  $\mathcal{U}$ ;
4    $\hat{\mu}, \hat{\sigma}, \hat{\phi} \leftarrow$  get data map statistics of  $\mathcal{L}$  with  $\theta$ ;
5    $P_{\theta'} \leftarrow$  train binary classifier  $\theta'$  on  $\Psi(\mathcal{L})$  with
   
$$y_{\Psi(\hat{x}_i)} = \begin{cases} 1, & \text{if } \hat{\phi}_i > t_{cor}; \\ 0, & \text{else} \end{cases}$$

6   for  $j = 1, \dots, \frac{K}{n}$  do
7      $\hat{x} \leftarrow \operatorname{argmin}_{\mathbf{x} \in \Psi(\mathcal{U})} |0.5 - P_{\theta'}(\hat{y} = 1 \mid \mathbf{x})|$ ;
8      $\mathcal{L} \leftarrow \mathcal{L} \cup \hat{x}$ ;
9      $\mathcal{U} \leftarrow \mathcal{U} \setminus \hat{x}$ ;
10  end
11  reset parameters  $\theta$  and  $\theta'$ ;
12 end
13 return  $\mathcal{L}, \mathcal{U}$ ;

```

---

## 4.4 Experimental Setup

We focus on pool-based active learning. Once trained on a seed set  $\mathcal{L}$ , we begin the simulated AL loop by iteratively selecting instances based on the scoring of an acquisition function. We take the top-50 instances, following prior work (Gissin and Shalev-Shwartz, 2019; Ein-Dor et al., 2020). The selected instances are shown the withheld label and added to the labeled set  $\mathcal{L}$  and removed from  $\mathcal{U}$ . We evaluate the performance of the trained model on a predefined held-out test set. We run 30 AL iterations, over five random seeds, and report averages over these runs.

### 4.4.1 Datasets

In our AL setup, we consider two popular text classification tasks, namely AGNews (Zhang et al., 2015) and TREC (Li and Roth, 2002). The AGNews

Dataset	Train	Test	Classes	Seed set size
AGNews	120,000	7,600	4	1,000
TREC	5,452	500	6	500

Table 4.1: **Datasets.** Statistics of the two datasets.

task entails classifying news articles into four classes: world, sports, business, science/technology. For TREC, the task is to categorize questions into one of six categories based on the subject of the question, such as questions about locations, persons, concepts, et cetera. Statistics of the data can be found in Table 4.1. We start with a seed set size of 1,000 for AGNews and 500 for TREC, both are stratified. This means after the AL iterations we will have 2,500 labeled instances for AGNews and 2,000 for TREC. Our motivation here is to keep the AL simulation realistic. We assume enough annotation budget to initially annotate 500–1,000 samples. Then, in every AL iteration annotate an additional 50 samples, which seems manageable for an annotator. Finally, we run 30 AL iterations to give a good overview of the performance of the acquisition functions over the iterations towards convergence.

#### 4.4.2 Acquisition Functions

We consider five acquisition functions. We opt for a random sampling baseline (**Rand.**), four existing acquisition functions, and our proposed CAL algorithm. We chose these as they are state-of-the-art and cover a spectrum of acquisition functions (uncertainty, batch-mode and diversity-based).

**Least Confidence (LC; Culotta and McCallum, 2005).** It takes

$$\operatorname{argmax}_{\boldsymbol{x} \in \mathcal{U}} 1 - P_{\theta}(\hat{y} \mid \boldsymbol{x})$$

of the predictive (e.g. softmax) distribution as the model’s uncertainty, and chooses instances with lowest predicted probability.

**Max-Entropy (Ent.; Dagan and Engelson, 1995).** Another popular example is entropy based sampling. Instances are selected according to the function

$$\operatorname{argmax}_{\mathbf{x} \in \mathcal{U}} - \sum_{y \in \mathcal{Y}} P_{\theta}(y | \mathbf{x}) \log_2 (P_{\theta}(y | \mathbf{x}))$$

and again, based on the *a posteriori* probability distribution.

**Bayesian Active Learning by Disagreement (BALD; Houlsby et al., 2011; Gal and Ghahramani, 2016).** This approach entails applying dropout at test time, then estimating uncertainty as the disagreement between outputs realized via multiple passes through the model. We use the Monte Carlo Dropout technique on ten inference cycles, with the max-entropy acquisition function.

**Discriminative Active Learning (DAL; Gissin and Shalev-Shwartz, 2019).** This approach poses AL as a binary classification task, it uses a separate binary classifier as a proxy to select instances that make  $\mathcal{L}$  representative of the entire dataset (i.e., making the labeled set indistinguishable from the unlabeled pool set). The input space for the binary classifier is task-agnostic. One maps the original input space  $\mathcal{X}$  to a learned representation  $\hat{\mathcal{X}}$  as the input space, with label space  $\mathcal{Y} = \{l, u\}$  referring to labeled and unlabeled. In the original paper, the learned representation is defined as the logits of the last hidden layer of the main classifier which solves the original task. Formally, it selects the top- $k$  instances that satisfy

$$\operatorname{argmax}_{\mathbf{x} \in \mathcal{U}} \hat{P}_{\theta}(\hat{y} = u | \Psi(\mathbf{x}))$$

where  $\hat{P}_{\theta}$  is the trained binary classifier given the learned representations  $\Psi$  of instances  $\mathbf{x}$ .



### 4.4.3 Configurations

This work uses two models for the AL setup solving the classification tasks. All the code is open source and available to reproduce our results.<sup>3</sup>

**Main Classifier** We use a Multi-layer Perceptron (MLP), with three  $d_{\text{emb}} = 300$  ReLu layers, dropout probability  $p = 0.3$ , minimize cross-entropy, AdamW optimizer (Loshchilov and Hutter, 2019), with a learning rate of  $1e-4$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $\epsilon = 1e-8$ .

**Binary Classifier** This model is suited for the binary classification task of DAL and CAL. In this case it is a single  $d_{\text{emb}} = 300$  ReLu layer. We minimize the cross-entropy as well. We use the AdamW optimizer with a learning rate of  $5e-5$ . For more details regarding reproducibility, we refer to Subsection 4.7.1.

**Training the Binary Classifier** For both the binary classification task of DAL and CAL, we empirically determined that setting the number of epochs for the binary classifier to 30 yielded good results. In the context of DAL, for AGNews it reaches around 98% accuracy during training, and for TREC it achieves around 89% accuracy. In contrast, with CAL, we start with little amounts of data for a binary classifier to train on. In the early stage of the AL iterations, the binary classifier does not achieve a high accuracy for both AGNews and TREC (around random). After it reaches the fifth or sixth AL iteration it starts to properly distinguish the **low-cor/high-cor** samples, as it probably has enough samples to learn from. It achieves around 65–75% accuracy for AGNews, and towards 85% accuracy for TREC. The classification accuracy on AGNews seems low. However, further tuning of the binary classifier (e.g., increasing the number of epochs) slightly increases binary classification accuracy, but did not result in better performance for the overall AL setup.

---

<sup>3</sup><https://github.com/jjzha/cal>

**Significance** Recently, the Almost Stochastic Order test (ASO; Dror et al., 2019)<sup>4</sup> has been proposed to test statistical significance for DNNs over multiple runs. Generally, the ASO test determines whether a stochastic order (Reimers and Gurevych, 2018) exists between two models or algorithms based on their respective sets of evaluation scores. Given the single model scores over multiple random seeds of two algorithms  $A$  and  $B$ , the method computes a test-specific value ( $\epsilon_{min}$ ) that indicates how far algorithm  $A$  is from being significantly better than algorithm  $B$ . When distance  $\epsilon_{min} = 0.0$ , one can claim that  $A$  stochastically dominant over  $B$  with a predefined significance level. When  $\epsilon_{min} < 0.5$  one can say  $A \succeq B$ . On the contrary, when we have  $\epsilon_{min} = 1.0$ , this means  $B \succeq A$ . For  $\epsilon_{min} = 0.5$ , no order can be determined. We took 0.05 for the predefined significance level  $\alpha$  and measure it across all 30 AL iterations.

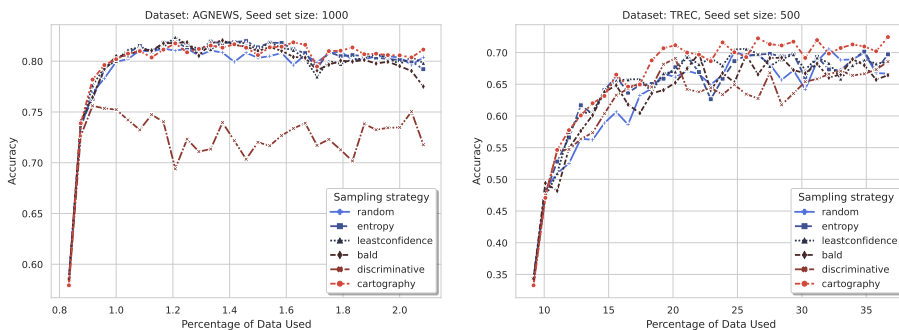


Figure 4.3: **Performance AL strategies.** Performance of the various AL strategies in terms of accuracy. The accuracy shown over the AL iterations is the average over five random seeds. Note that for both datasets we added the same number of instances to the seed set (+1,500 instances). The x-axis correspond to the fraction of the total size of the respective dataset.

<sup>4</sup>Implementation of Dror et al. (2019) can be found at <https://github.com/Kaleidophon/deep-significance> (Ulmer, 2021)

	Rand.	LC	Ent.	BALD	DAL	CAL
Rand.		1.00	1.00	1.00	0.01	1.00
LC	<b>0.00</b>		1.00	1.00	0.01	1.00
Ent.	<b>0.00</b>	<b>0.00</b>		1.00	0.01	1.00
BALD	<b>0.00</b>	<b>0.00</b>	0.00		0.01	1.00
DAL	0.99	0.99	0.99	0.99		1.00
CAL	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	

	Rand.	LC	Ent.	BALD	DAL	CAL
Rand.		1.00	1.00	1.00	1.00	1.00
LC	<b>0.00</b>		1.00	0.78	0.57	1.00
Ent.	<b>0.00</b>	<b>0.00</b>		0.87	0.60	1.00
BALD	<b>0.00</b>	0.22	0.13		1.00	1.00
DAL	<b>0.00</b>	0.43	0.40	1.00		1.00
CAL	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	

Table 4.2: **Almost Stochastic Order Scores of AGNews (left) & TREC (right)**. ASO scores expressed in  $\epsilon_{min}$ . The significance level  $\alpha = 0.05$  is adjusted accordingly by using the Bonferroni correction (Bonferroni, 1936). **Bold** numbers indicate stochastic dominance and *cursive* means that one algorithm is better than the other, e.g., for AGNews, LC (row) is stochastically dominant over the random baseline (column) with  $\epsilon_{min}$  value of 0.00). Numbers are obtained across all 30 AL iterations

## 4.5 Results & Analysis

We plot the accuracy of the AL algorithms (Subsection 4.4.2) on each dataset in Figure 4.3. For AGNews and TREC, all AL strategies except DAL outperform the random baseline. CAL is statistically dominant over BALD (AGNews) and DAL (AGNews, TREC), and competitive with LC and Entropy (Table 4.2). This shows that CAL reaches strong results. CAL (illustrated as cartography in the figure with a red-dotted line).

### 4.5.1 Why does CAL work?

To gain insight on why CAL works better, we investigate the average statistics of each selected batch of samples using the data maps. In Figure 4.4, we check the mean confidence, variability and correctness over each selected batch of 50 for sampling strategies Random, LC, DAL, and CAL for both AGNews and TREC. The statistics of the instances are extracted after the selected top-50 batch is added to the seed set. Once the main model is trained again on the increased seed set, we obtain the statistics of the previously added batch of 50. Figure 4.4 shows that leastconfidence and cartography has the higher variability amongst the other methods (middle graph). There is a similar signal as the findings of Swayamdipta

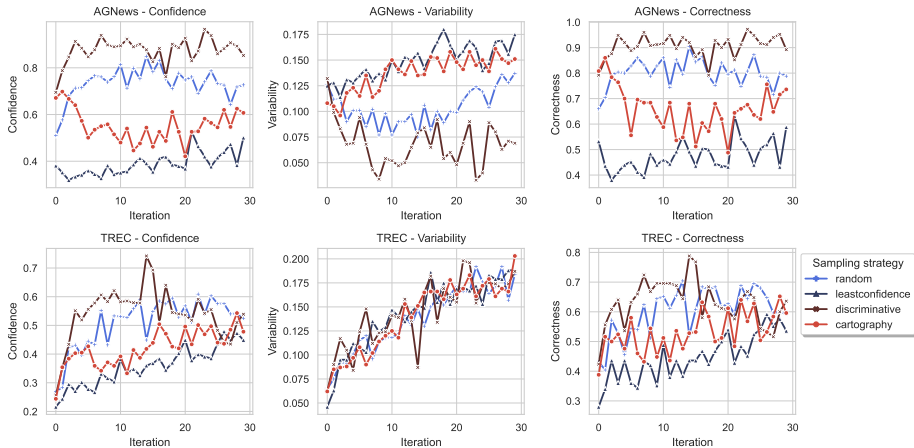


Figure 4.4: **Average Statistics AGNews & TREC.** Values correspond to the mean statistics (Section 4.3) of instances over the five random seeds. We calculate the statistics of each top-50 batch *after* being added to the seed set. Therefore, we only have statistics of 29 runs, as in the 30th run we stop the AL cycle.

et al. (2020a), the *ambiguous* samples that the model *learns the most* from are usually the instances that have the highest variability and average confidence—as we can see in Figure 4.4 (especially pronounced on AGNews), these are the instances CAL selects. In contrast, LC selects instances that have relatively low confidence and low variability in the early stages, but catches up in later ones. This suggests that LC chooses only *hard-to-learn* instances at the start. In general, CAL seems to select more *informative* samples as opposed to the random strategy.

DAL seems to start by choosing *ambiguous* samples with high variability. However, it quickly picks mostly **high-cor** samples in later iterations, in contrast to CAL. Consequently, the performance for DAL drops as it leads to picking the *easy-to-learn* samples. Picking too many *easy-to-learn* instances results in worse optimization (Swayamdipta et al., 2020a). The drop for DAL is visible in Figure 4.3, which shows that CAL and DAL are close at start, and the accuracy of DAL then drops.

### 4.5.2 What is the influence of the *correctness* threshold?

$t_{cor}$	0.0	<b>≤ 0.2</b>	≤ 0.4	≤ 0.6	≤ 0.8
AGNews	0.789	<b>0.811</b>	0.796	0.788	0.781
TREC	0.676	<b>0.724</b>	0.701	0.706	–

Table 4.3: **Influence of the Correctness Threshold.** Influence of the correctness threshold  $t_{cor}$  on the final average accuracy score per dataset over the five random seeds.  $t_{cor}$  considers from what boundary we should consider **low-cor** cases. In **bold** is what threshold we are using.

Here we investigate how changing the correctness threshold for the binary classification task impacts accuracy. As shown in Table 4.3, the selected threshold worked well for the two datasets studied. In particular, the accuracy does not drop substantially on AGNews if we move the *correctness* threshold to a higher value. Swayamdipta et al. (2020a) indicates that the *ambiguous* region contains the instances with the highest variability. For AGNews, these instances have a *correctness* range from 0.2–0.8 (as seen in Figure 4.1). Therefore, we assume that most of these instances are informative for the model. In the case of TREC, the final accuracy also stays similar with different *correctness* thresholds and does not drop substantially. We find in the full data map for TREC that the area with  $t_{cor} = \{0.0, 0.4\}$  thresholds is more dense compared to  $t_{cor} = 0.2$  (details in Subsection 4.7.2). In other words, there are more samples, but also ones that are detrimental to performance. We note that the *correctness* threshold could vary for different datasets and models.

### 4.5.3 Do AL strategies select the same instances for labeling?

We measured the overlap between the batches selected by each pair of strategies (Random, LC, DAL, CAL) on AGNews and TREC (Table 4.4). The batch overlap for CAL is low, with the highest overlap being 7 instances for AGNews with  $CAL \cap Rand.$  followed by  $CAL \cap DAL$  with 4 overlapping instances. The highest overlap for TREC is 114 instances for  $CAL \cap DAL.$

	AGNews	TREC
CAL $\cap$ DAL	4	114
CAL $\cap$ LC	2	101
CAL $\cap$ Rand.	7	82
DAL $\cap$ LC	3	81
DAL $\cap$ Rand.	2	87
LC $\cap$ Rand.	2	100

Table 4.4: **Number of Overlapping Instances on AGNews & TREC.** The values corresponds to the total overlapping instances out of  $1,500 * 5$  random seeds = 7,500.

Note this is the total overlap over five seeds. These results indicate that the AL algorithms choose different instances.

A popular approach for improving classification performance is combining (complementary) AL strategies. For example, [Zhdanov \(2019\)](#) proposed the idea to combine uncertainty sampling and diversity sampling for image classification. As DAL and CAL have few overlapping instances, they can be complementary to each other. To test this, we combined DAL and CAL using a simple heuristic, by providing both of them half of the annotation budget (i.e., take top-25 batch of each AL strategy). This resulted in an accuracy score of 0.695 for TREC and 0.789 for AGNews. This suggests that it could have a positive effect if there is a more sophisticated approach. This is an open research topic that requires further investigation.

#### 4.5.4 How data-efficient is CAL in comparison to full data training?

If a model is able to choose the instances that it can learn the most from, it can reach comparable results or even outperform a model trained on all data by using fewer training instances. This is noted by [Siddhant and Lipton \(2018\)](#), where they achieve 98–99% of the full dataset performance while labeling only 20% of the samples. The overall accuracy for AGNews trained on all data is 0.820 accuracy on test. For TREC, this results in 0.634.

With CAL, we achieve around 99% of the full dataset performance while using only 2% training data for AGNews. For TREC, we outperform the full dataset performance by 0.090 accuracy (0.634 vs. 0.724), the full dataset performance is already reached by using around 14% training data. This is appealing, as active learning can provide more data-effective learning solutions.

## 4.6 Conclusion

In this paper, we introduced a new AL algorithm, Cartography Active Learning. The AL objective is transformed into a binary classification task (Gissin and Shalev-Shwartz, 2019), where we optimize for selecting the most informative data with respect to a model by leveraging insights from data maps (Swayamdipta et al., 2020a). Data maps help to identify distinct regions in a dataset based on training dynamics (*hard-to-learn* and *easy-to-learn/ambiguous* instances), which have shown to play an important role in model optimization and stability in full dataset training (Swayamdipta et al., 2020a). We use these insights in low-data regimes and propose CAL. Specifically for the task of Visual Question Answering, contemporary work by Karamcheti et al. (2021) use data maps for analysis into AL acquisition functions and the effect of outliers. In contrast, this work instead applies data maps directly for AL by training a discriminator on limited seed data maps to select the most informative instances. We show empirically that our method is competitive or significantly outperforms various popular AL methods, and provide intuitions on why this is the case by using training dynamics.

## 4.7 Appendix

### 4.7.1 Reproducibility

We initialize the main model with English  $d_{\text{emb}} = 300$  FastText embeddings (Bojanowski et al., 2017) and keep it frozen during training and inference, we sum the embeddings over the sequence of tokens as motivated by Banea et al. (2014). For AGNews we impose a maximum sequence length of 200 and a batch size of 64. For TREC, a maximum sentence length of 42 and a batch size of 16. We run both models for 30 epochs, with no early stopping. Per AL iteration, we do a weight reset on all models. We average our results over five randomly generated seeds (398048, 127003, 259479, 869323, 570852). All experiments were ran on an NVIDIA<sup>®</sup> A100 SXM4 40 GB GPU and an AMD EPYC<sup>™</sup> 7662 64-Core CPU. Specifically for CAL, a single AL batch (50) selection iteration takes 11 seconds on average assuming TREC. For AGNews, one AL iteration takes 62 seconds on average. Both runtimes are with respect to the models depicted in Subsection 4.4.3 and hardware mentioned above.

### 4.7.2 Full Data Map

Figure 4.5 and Figure 4.6 show the full data maps for AGNews and TREC respectively. Identically to Swayamdipta et al. (2020a), we show the density of data points in the plots. We can see a clear difference in density between the datasets. For AGNews, we can see the majority of data points have a high confidence ( $\sim 0.8$ ) and high correctness. In contrast, TREC contains plenty of instances that have low confidence ( $\sim 0.3$ ) and low correctness.



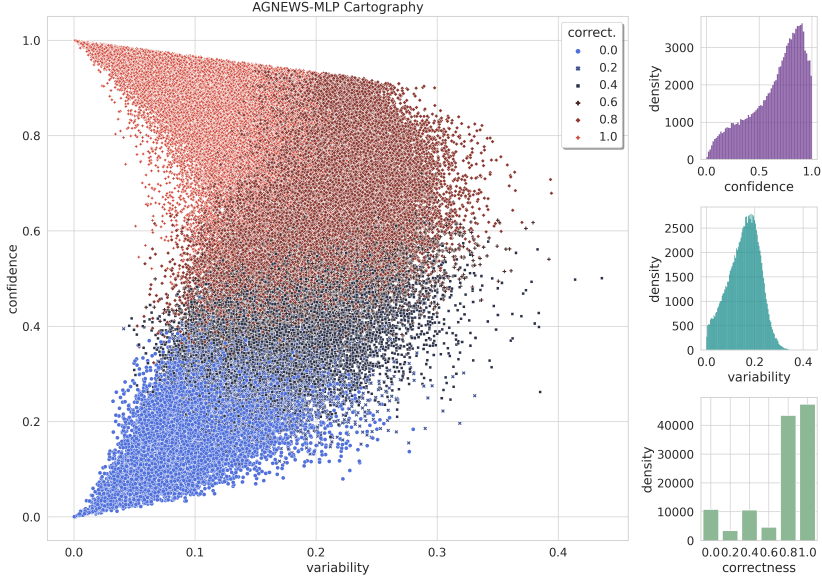


Figure 4.5: **Density of AGNews.** Density statistics of AGNews over ten epochs.

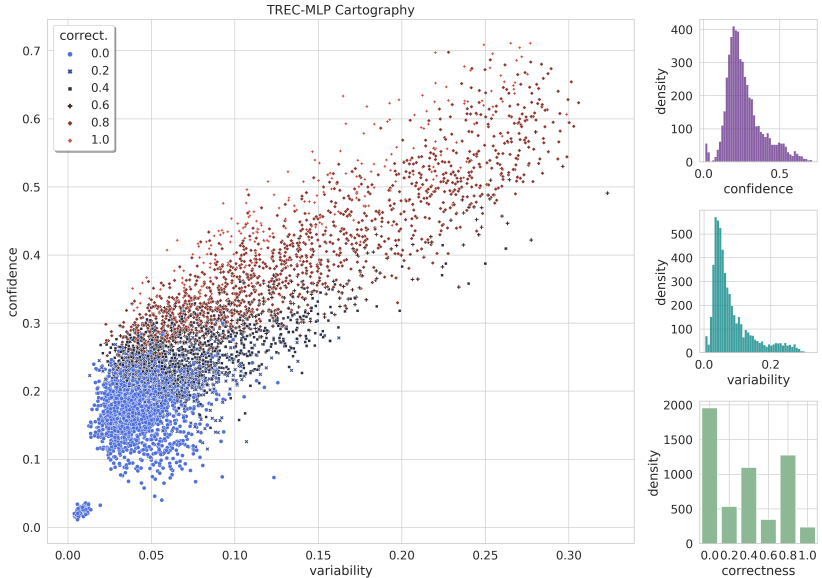


Figure 4.6: **Density of TREC.** Density statistics of TREC over ten epochs.



# Chapter 5

## SKILLSPAN: Hard and Soft Skill Extraction from English Job Postings

The work presented in this chapter is based on a paper that has been published as: Mike Zhang, Kristian Jensen, Sif Sonniks, and Barbara Plank. SkillSpan: Hard and soft skill extraction from English job postings. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4962–4984, Seattle, United States, July 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.366. URL <https://aclanthology.org/2022.naacl-main.366>.

## Abstract

Skill Extraction (SE) is an important and widely-studied task useful to gain insights into labor market dynamics. However, there is a lacuna of datasets and annotation guidelines; available datasets are few and contain crowd-sourced labels on the span-level or labels from a predefined skill inventory. To address this gap, we introduce SKILLSPAN, a novel SE dataset consisting of 14.5K sentences and over 12.5K annotated spans. We release its respective guidelines created over three different sources annotated for hard *and* soft skills by domain experts. We introduce a BERT baseline (Devlin et al., 2019b). To improve upon this baseline, we experiment with language models that are optimized for long spans (Joshi et al., 2020a; Beltagy et al., 2020), continuous pre-training on the job posting domain (Han and Eisenstein, 2019; Gururangan et al., 2020a), and multi-task learning (Caruana, 1997). Our results show that the domain-adapted models significantly outperform their non-adapted counterparts, and single-task outperforms multi-task learning.

## 5.1 Introduction

Job markets are under constant development—often due to developments in technology, migration, and digitization—so are the skill sets required. Consequentially, job vacancy data is emerging on a variety of platforms in big quantities and can provide insights on labor market skill demands or aid job matching (Balog et al., 2012). SE is to extract the competences necessary from unstructured text.

Previous work in SE shows promising progress, but is halted by a lack of available datasets and annotation guidelines. Two out of 14 studies release their dataset, which limit themselves to crowd-sourced labels (Sayfullina et al., 2018) or annotations from a predefined list of skills on the document-level (Bhola et al., 2020). Additionally, none of the 14 previously mentioned studies release their annotation guidelines, which obscures the meaning of a competence. Job markets change, as do the

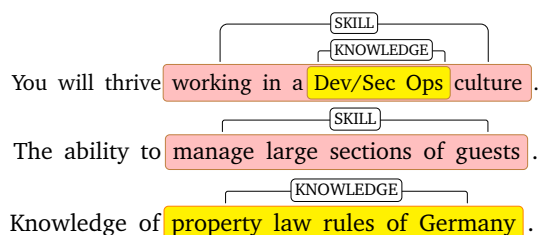


Figure 5.1: **Examples of Skills & Knowledge Components.** Annotated samples of passages in varying job postings. More details are given in Section 5.4.

skills in, e.g., the European Skills, Competences, Qualifications and Occupations (ESCO; le Vrang et al., 2014) taxonomy (Section 5.3). Hence, it is important to cover for possible emerging skills.

We propose SKILLSPAN, a novel SE dataset annotated at the span-level for *skill* and *knowledge* components (SKCs) in job postings (JPs). As illustrated in Figure 5.1, SKCs can be nested inside skills. SKILLSPAN allows for extracting possibly undiscovered competences and to diminish the lack of coverage of predefined skill inventories.

Our analysis (Figure 5.2) shows that SKCs contain on average longer sequences than typical Named Entity Recognition (NER) tasks. Albeit we additionally study models optimized for long spans (Joshi et al., 2020b; Beltagy et al., 2020), some underperform. Overall, we find specialized domain BERT models (Alsentzer et al., 2019a; Lee et al., 2020a; Gururangan et al., 2020b; Nguyen et al., 2020a) perform better than their non-adapted counterparts. We explore the benefits of domain-adaptive pre-training on the JP domain (Han and Eisenstein, 2019; Gururangan et al., 2020b). Last, given the examples from Figure 5.1, we formulate the task as both as a sequence labeling and a multi-task learning (MTL) problem, i.e., training on both skill and knowledge components jointly (Caruana, 1997).

**Contributions.** In this paper:



1. We release SKILLSPAN, a novel skill extraction dataset, with annota-



tion guidelines, and our open-source code.<sup>1</sup>

2. We present strong baselines for the task including a new SpanBERT (Joshi et al., 2020b) trained from scratch, and domain-adapted variants (Gururangan et al., 2020b), which we will release on the HuggingFace platform (Wolf et al., 2020a). To the best of our knowledge, we are the first to investigate the extraction of skills and knowledge from job postings with state-of-the-art language models.
3. We give an analysis of single-task versus multi-task learning in the context of skill extraction, and show that for this particular task single-task learning outperforms multi-task learning.

---

<sup>1</sup><https://github.com/kris927b/SkillSpan>

Paper	Annotations	Approach	Size	Skill Type	(Baseline) Model(s)		
Kivimäki et al. (2013)	Document-level	Automatic	N/A	Hard	LogEnt., TF-IDF, LSA, LDA	X	X
Zhao et al. (2015)	Sentence-level	Automatic	N/A	Hard	Word2Vec	X	X
Javed et al. (2017)	Span-level	Skill Inventory	N/A	Both	Word2Vec	X	X
Jia et al. (2018)	Span-level	Automatic	21,158 JPs*	Hard	LSTM	X	X
Sayfullina et al. (2018)	Span-level	Crowdsourcing	4,863 Sent.	Soft	CNN, LSTM	✓	X
Smith et al. (2019)	Span-level	Manual	100 JPs	Hard	Pattern Matching	X	X
Gugnani and Misra (2020)	Span-level	Domain Experts	~200 JPs	Hard	Word2Vec, Doc2Vec	X	X
Li et al. (2020d)	Document-level	Proprietary	N/A	Hard	FastText	X	X
Shi et al. (2020)	Span-level	Proprietary	N/A	Hard	FastText, USE, BERT	X	X
Tamburri et al. (2020)	Sentence-level	Domain Experts	~3,000 Sent.	Both	BERT	X	X
Chernova (2020)	Span-level	Manual	100 JPs	Both	FinBERT	X	X
Bhola et al. (2020)	Document-level	Skill Inventory	20,298 JPs*	Hard	BERT	✓	X
Smith et al. (2021)	Span-level	Manual	100 JPs	Hard	Pattern Match., Word2Vec	X	X
Liu et al. (2021)	Document-level	Crowdsourcing	N/A	Hard	GNN	X	X
<b>This work</b>	Span-level	Domain Experts	391 JPs	Both	(Domain-adapted) BERT	✓	✓

**Table 5.1: Contributions of Related Work.** We list the recent works of Skill Extraction. Note that (\*) indicates labels that are automatically inferred from some source (e.g., a predefined skill inventory) and not *manually* annotated. With respect to the annotation approach, “Manual” indicates uncertainty whether they used domain experts or not. Also note that many works do not release their dataset with annotations () nor guidelines (). The list is inspired by Khaouja et al. (2021).

## 5.2 Related Work

There is a pool of prior work relating to SE. We summarize it in [Table 5.1](#), depicting state-of-the-art approaches, level of annotations, what kind of competences are annotated, the modeling approaches, the size of the dataset (if available), type of skills annotated for, baseline models, and whether they release their annotations and guidelines.

As can be seen in [Table 5.1](#), many works do *not release their data* (apart from [Sayfullina et al., 2018](#) and [Bhola et al., 2020](#)) and **none release their annotation guidelines**. In addition, none of the previous studies approach SE as a span-level extraction task with state-of-the-art language models, nor did they release a dataset of this magnitude with manually annotated (long) spans of competences by domain experts.

Although [Sayfullina et al. \(2018\)](#) annotated on the span-level (thus being useful for SE) and release their data, they instead explored several approaches to *Skill Classification*. To create the data, they extracted all text snippets containing *one* soft skill from a predetermined list. Crowdworkers then annotated the highlighted skill whether it was a soft skill referring to the candidate or not. They show that an LSTM ([Hochreiter et al., 1997](#)) performs best on classifying the skill in the sentence. In our work, we annotated a dataset three times their size ([Table 5.2](#)) for both hard *and* soft skills. In addition, we also extract the specific skills from the sentence.

[Tamburri et al. \(2020\)](#) classifies sentences that contain skills in the JP. The authors manually labeled their dataset with domain experts. They annotated whether a sentence contains a skill or not. Once the sentence is identified as containing a skill, the skill cited within is extracted. In contrast, we directly annotate for the span within the sentence.

[Bhola et al. \(2020\)](#) cast the task of skill extraction as a multi-label skill classification at the document-level. There is a predefined set of unique skills given the job descriptions and they predict multiple skills that are connected to a given job description using BERT ([Devlin et al., 2019b](#)). In addition, they experiment with several additional layers for better prediction performance. We instead explore domain-adaptive pre-



training for SE.

The work closest to ours is by Chernova (2020), who approach the task similarly with span-level annotations (including longer spans) but approach this for the Finnish language. It is unclear whether they annotated by domain experts. Also, neither the data nor the annotation guidelines are released. For a comprehensive overview with respect to SE, we refer to Khaouja et al. (2021).

### 5.3 Skill & Knowledge Definition

There is an abundance of competences and there have been large efforts to categorize them. For example, the The International Standard Classification of Occupations (ISCO; Elias, 1997) is one of the main international classifications of occupations and skills. It belongs to the international family of economic and social classifications. Another example, the European Skills, Competences, Qualifications and Occupations (ESCO; le Vrang et al., 2014) taxonomy is the European standard terminology linking skills and competences and qualifications to occupations and derived from ISCO. The ESCO taxonomy mentions three categories of competences: *Skill*, *knowledge*, and *attitudes*. ESCO defines knowledge as follows:

“Knowledge means the outcome of the assimilation of information through learning. Knowledge is the body of facts, principles, theories and practices that is related to a field of work or study.”<sup>2</sup>

For example, a person can acquire the Python programming language through learning. This is denoted as a *knowledge* component and can be considered a *hard skill*. However, one also needs to be able to apply the knowledge component to a certain task. This is known as a *skill* component. ESCO formulates it as:

---

<sup>2</sup><https://ec.europa.eu/esco/portal/escopedia/Knowledge>

“Skill means the ability to apply knowledge and use know-how to complete tasks and solve problems.”<sup>3</sup>

In ESCO, the *soft skills* are referred to as *attitudes*. ESCO considers attitudes as skill components:

“The ability to use knowledge, skills and personal, social and/or methodological abilities, in work or study situations and professional and personal development.”<sup>4</sup>

To sum up, hard skills are usually referred to as *knowledge* components, and applying these hard skills to something is considered a *skill* component. Then, soft skills are referred to as *attitudes*, these are part of skill components. There has been no work, to the best of our knowledge, in annotating skill and knowledge components in JPs.

---

<sup>3</sup><https://ec.europa.eu/esco/portal/escopedia/Skill>

<sup>4</sup><http://data.europa.eu/esco/skill/A>

	↓ Statistics, Src. →	BIG	HOUSE	TECH	Total
Train	# Posts	60	60	80	200
	# Sentences	1,036	1,674	3,156	5,866
	# Tokens	29,064	36,995	56,549	122,608
	# Skill Spans	1,086	984	1,237	3,307
	# Knowledge Spans	439	781	2,188	3,408
	# Overlapping Spans	45	29	135	209
Development	# Posts	30	30	30	90
	# Sentences	783	1,022	2,187	3,992
	# Tokens	11,762	19,173	21,149	52,084
	# Skill Spans	469	525	545	1,539
	# Knowledge Spans	126	287	806	1,219
	# Overlapping Spans	12	17	32	61
Test	# Posts	36	33	32	101
	# Sentences	1,112	1,216	2,352	4,680
	# Tokens	14,720	21,923	20,885	57,528
	# Skill Spans	634	637	459	1,730
	# Knowledge Spans	242	350	834	1,426
	# Overlapping Spans	12	8	9	29
Total	# Posts	126	123	142	391
	# Sentences	2,931	3,912	7,695	14,538
	# Tokens	55,546	78,091	98,583	232,220
	# Skill Spans	2,189	2,146	2,241	6,576
	# Knowledge Spans	807	1,418	3,828	6,053
	# Overlapping Spans	69	54	178	301
$\mathcal{U}$	# Posts				126,769
	# Sentences				3,195,585
	# Tokens				460,484,670

Table 5.2: **Statistics of Dataset.** Indicated is the number of JPs across splits & source and their respective number of sentences, tokens, and spans. The total is reported in the cyan column and rows. We report the overall statistics of the unlabeled JPs ( $\mathcal{U}$ ) in the gray rows.

## 5.4 SKILLSPAN Dataset

### 5.4.1 Data

We continuously collected JPs via web data extraction between June 2020–September 2021.<sup>5</sup> Our JPs come from three sources:

1. **BIG**: A large job platform with various types of JPs, with several type of positions;
2. **HOUSE**: A *static* in-house dataset consisting of similar types of jobs as BIG. Dates range from 2012–2020;
3. **TECH**: The StackOverflow JP platform that consisted mostly of technical jobs (e.g., developer positions).

We release the anonymized raw data and annotations of the parts with permissible licenses, i.e., HOUSE (from a governmental agency which is our collaborator) and TECH.<sup>6</sup> For anonymization, we perform it via manual annotation of job-related sensitive and personal data regarding Organization, Location, Contact, and Name following the work by Jensen et al. (2021a).

Table 5.2 shows the statistics of SKILLSPAN, with 391 annotated JPs from the three sources containing 14.5K sentences and 232.2K tokens. The unlabeled JPs (only to be released as pre-trained model) consist of 126.8K posts, 3.2M sentences, and 460.5M tokens. What stands out is that there are 2–5 times as many annotated knowledge components in TECH in contrast to the other sources, despite a similar amount of JPs. We expect this to be due the numerous KCs depicted in this domain (e.g., programming languages), while we observe considerably fewer soft skills (e.g., “work flexibly”). The amount of skills is more balanced across the three sources. Furthermore, overlapping spans follow a consistent trend among splits, with the train split containing the most.

---

<sup>5</sup>Our data statement (Bender and Friedman, 2018) can be found in Subsection 5.9.1

<sup>6</sup>Links to our data can be found at <https://github.com/kris927b/SkillSpan>.

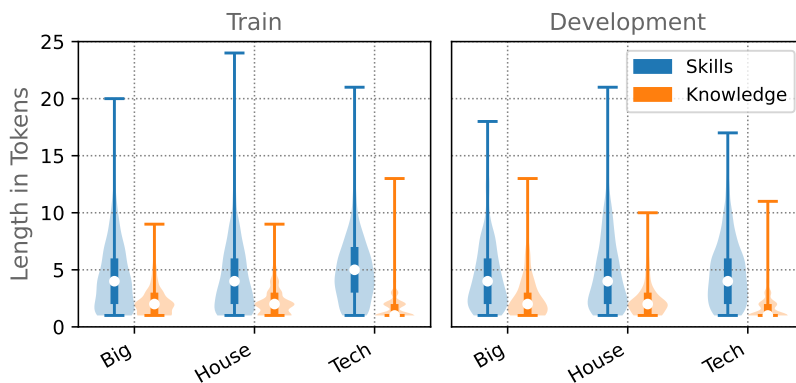


Figure 5.2: **Violin Plots of Annotated Components.** Indicated are the distributions regarding the length of spans in each type of annotated component (i.e., length of skills and knowledge components). The white dot is the median length, the bars range from the first quartile to the third quartile, and the colored line ranges from the lower adjacent value to the higher adjacent value.

### 5.4.2 Data Annotation

We annotate competences related to SKCs in two levels as illustrated in Figure 5.1. We started the process in March 2021, with initial annotation rounds to construct and refine the annotation guidelines (as outlined further below). The annotation process spanned eight months in total. Our final annotation guidelines can be found in Subsection 5.9.2. The guidelines were developed by largely following example spans given in the ESCO taxonomy. However, at this stage, we focus on span identification, and we do not take the fine-grained taxonomy codes from ESCO for labeling the spans, leaving the mapping to ESCO and taxonomy enrichment as future work.

### 5.4.3 Further Details on the Annotation Process

The development of the annotation guidelines and our annotation process is depicted as follows: ① We wrote base guidelines derived from a small

number of JPs. ② We had three pre-rounds consisting of three JPs each. After each round, we modified, improved and finalized the guidelines. ③ Then, we had three longer-lasting annotation rounds consisting of 30 JPs each. We re-annotated the previous 11 JPs in ① and ②. ④ After these rounds, one of the annotators (the hired linguist) annotated JPs in batches of 50. The data in ①, ②, and ③ was annotated by three annotators (101 JPs).

We used an open source text annotation tool named DOCCANO (Nakayama et al., 2018). There are around 57.5K tokens (approximately 4.6K sentences, in 101 job posts) that we calculated agreement on. The annotations were compared using Cohen’s  $\kappa$  (Fleiss and Cohen, 1973) between pairs of annotators, and Fleiss’  $\kappa$  (Fleiss, 1971), which generalises Cohen’s  $\kappa$  to more than two concurrent annotations. We consider two levels of  $\kappa$  calculations: **TOKEN** is calculated on the token level, comparing the agreement of annotators on each token (including non-entities) in the annotated dataset. **SPAN** refers to the agreement between annotators on the exact span match over the surface string, regardless of the type of SKC, i.e., we only check the position of tag without regarding the type of the entity. The observed agreements scores over the three annotators from step ③ are between 0.70–0.75 Fleiss’  $\kappa$  for both levels of calculation which is considered a *substantial agreement* (Landis and Koch, 1977) and a  $\kappa$  value greater than 0.81 indicates *almost perfect agreement*. Given the difficulty of this task, we consider the aforementioned  $\kappa$  score to be strong. Particularly, we observed a large improvement in annotation agreement from the earlier rounds (step ① and ②), where our Fleiss’  $\kappa$  was 0.59 on token-level and 0.62 for the span-level.

Overall, we observe higher annotator agreement for knowledge components (3–5% higher) compared to skills which tend to be longer. The TECH domain is the most consistent for agreement while BIG shows more variation over rounds, likely due to the broader nature of the domains of JPs.

### 5.4.4 Annotation Span Statistics

A challenge of annotating spans is the length (i.e., boundary), SKCs being in different domains (e.g., business versus technical components), and frequently written differently, e.g., “being able to work together” v.s. “team-work”). [Figure 5.2](#) shows the statistics of our annotations in violin plots. For the training set, the median length (white dot) of skills is around 4 for BIG and HOUSE, for TECH this is a median of 5. In the development set, the median stays at length 4 across all sources. Another notable statistic is the upper and lower percentile of the length of skills and knowledge, indicated with the thick bars. Here, we highlight the fact that skill components could consist of many tokens, for example, up to length 7 in the HOUSE source split (see blue-colored violins). For knowledge components, the spans are usually shorter, where it is consistently below 5 tokens (see orange-colored violins). All statistics follow a similar distribution across train, development, and sources in terms of length and distribution. This gives a further strong indication that consistent annotation length has been conducted across splits and sources.

	BIG	HOUSE	TECH
SKILL	ambitious	structured	hands-on
	proactive	teaching	communication skills
	work independently	communication skills	leadership
	attention to detail	project management	passionate
	motivated	drive	open-minded
KNOWLEDGE	full uk driving licence	english	java
	sap energy assessments	supply chain	javascript
	right to work in the uk	project management	aws
	sen	powders	docker
	acca/aca	machine learning	node.js

Table 5.3: **Most Frequent Skills in the Development Data.** Top-5 skill components in our data in terms of frequency on different sources. A larger example can be found in [Table 5.8](#) and [Table 5.9](#) ([Subsection 5.9.5](#)).

### 5.4.5 Qualitative Analysis of Annotations

Qualitative differences in SKCs over the three sources are shown (lower-cased) in [Table 5.3](#). With respect to skill components, all sources follow a similar usage of skills. The annotated skills mostly relate to the *attitude* of a person and hence mostly consist of soft skills. With respect to knowledge components, we observe differences between the three sources. First, on the source-level, the knowledge components vastly differ between BIG and TECH. BIG postings seem to cover more *business* related components, whereas TECH has more *engineering* components. HOUSE seems to be a mix of the other two sources. Lastly, note that both the skill and knowledge components between the splits diverge in terms of the type of annotated spans, which indicates a variation in the annotated components. We show the top-10 skills annotated in the train, development, and test splits for SKCs in [Subsection 5.9.5](#). From a syntactic perspective, skills frequently consist of noun phrases, verb phrases, or adjectives (for soft skills). Knowledge components usually consists of nouns or proper nouns, such as “python”, “java”, and so forth.

## 5.5 Experimental Setup

The task of SE is formulated as a sequence labeling problem. Formally, we consider a set of JPs  $\mathcal{D}$ , where  $d \in \mathcal{D}$  is a set of sequences (i.e., entire JPs) with the  $i^{\text{th}}$  input sequence  $\mathcal{X}_d^i = \{x_1, x_2, \dots, x_T\}$  and a target sequence of BIO-labels  $\mathcal{Y}_d^i = \{y_1, y_2, \dots, y_T\}$  (e.g., “B-SKILL”, “I-KNOWLEDGE”, “O”). The goal is to use  $\mathcal{D}$  to train a sequence labeling algorithm  $h : \mathcal{X} \mapsto \mathcal{Y}$  to accurately predict entity spans by assigning an output label  $y_t$  to each token  $x_t$ .

As baseline we consider BERT and we investigate more recent variants, and we also train models from scratch. Models are chosen due to their state-of-the-art performance, or in particular, for their strong performance on longer spans.



### 5.5.1 BERT<sub>base</sub> (Devlin et al., 2019b)

An out-of-the-box BERT<sub>base</sub> model (bert-base-cased) from the HuggingFace library (Wolf et al., 2020a) functioning as a baseline.

### 5.5.2 SpanBERT (Joshi et al., 2020b)

A BERT-style model that focuses on span representations as opposed to single token representations. SpanBERT is trained by masking contiguous spans of tokens and optimizing two objectives: (1) masked language modeling, which predicts each masked token from its own vector representation. (2) The span boundary objective, which predicts each masked token from the representations of the unmasked tokens at the start and end of the masked span.

We train a SpanBERT<sub>base</sub> model from scratch on the BooksCorpus (Zhu et al., 2015) and English Wikipedia using cased Wordpiece tokens (Wu et al., 2016). We use AdamW (Kingma and Ba, 2015) for 2.4M training steps with batches of 256 sequences of length 512. The learning rate is warmed up for 10K steps to a maximum value of  $1e-4$ , after which it has a decoupled weight decay (Loshchilov and Hutter, 2019) of 0.1. We add a dropout rate of 0.1 across all layers. Pretraining was done on a v3-8 TPU on the GCP and took 14 days to complete. We take the official TensorFlow implementation of SpanBERT by Ram et al. (2021).

### 5.5.3 JobBERT

We apply domain-adaptive pre-training (Gururangan et al., 2020b) to a BERT<sub>base</sub> model using the 3.2M unlabeled JP sentences (Table 5.2).<sup>7</sup> Domain-adaptive pre-training relates to the continued self-supervised pre-training of a large language model on domain-specific text. This approach improves the modeling of text for downstream tasks within the domain. We continue training the BERT model for three epochs (default in HuggingFace) with a batch size of 16.

---

<sup>7</sup><https://huggingface.co/jjzha/jobbert-base-cased>

### 5.5.4 JobSpanBERT

We apply domain-adaptive pre-training to our SpanBERT on 3.2M unlabeled JP sentences.<sup>8</sup> We keep parameters identical to the vanilla SpanBERT, but change the number of steps to 40K to have three passes over the unlabeled data.

### 5.5.5 Experiments

We have 391 annotated JPs (Table 5.2) that we divide across three splits: Train, dev. and test set. We use 101 JPs that all three annotators annotated as the gold standard test set with aggregated annotations via majority voting. The 101 postings are divided between the sources as: 36 BIG, 33 HOUSE, and 32 TECH. The remaining 290 JPs were annotated by one annotator. We use 90 JPs (30 from each source, namely BIG, HOUSE, and TECH) as the dev. set. The remaining 200 JPs are used as the train set. The sources in the train set are divided into 60 BIG, 60 HOUSE, and 80 TECH.

### 5.5.6 Setup

The data is structured as CoNLL format (Tjong Kim Sang, 2002b). For the nested annotations, the skill tags are appearing only in the first column and the knowledge tags are only appearing in the second column of the file and they are allowed to overlap with each other. We perform experiments with single-task learning (STL) on either the skill or knowledge components, MTL for predicting both skill and knowledge tags at the same time, while evaluating the MTL models also on either skills or knowledge components. We used a single joint MTL model with hard-parameter sharing (Caruana, 1997). All models are with a final Conditional Random Field (CRF; Lafferty et al., 2001) layer. Earlier research, such as Souza et al. (2019); Jensen et al. (2021a) show that BERT models with a CRF-layer improve or perform similarly to its simpler variants when comparing the overall F1 and make no tagging errors (e.g., B-tag follows I-tag). In the case of MTL we use one

---

<sup>8</sup><https://huggingface.co/jjzha/jobspanbert-base-cased>

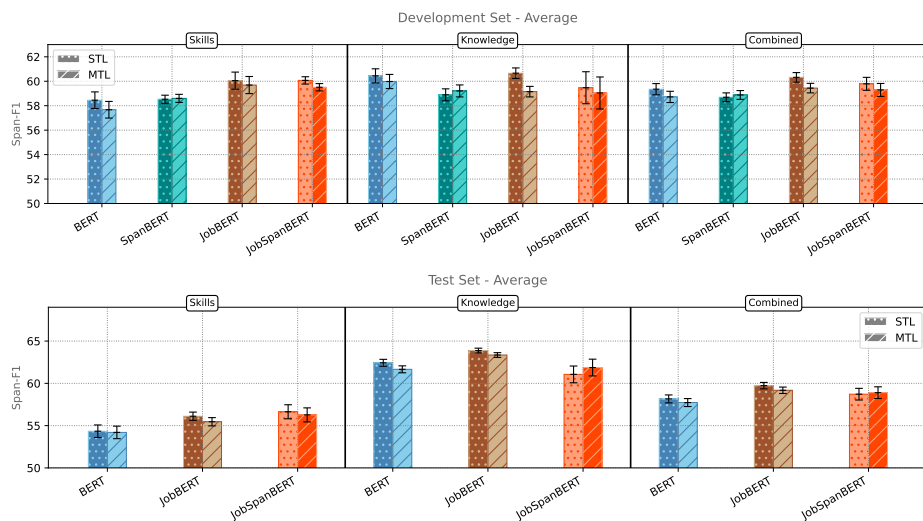


Figure 5.3: **Performance of Models.** We test the models on **SKILLS**, **KNOWLEDGE**, and **COMBINED**. We report the span-F1 and standard deviation (error bars) of runs on five random seeds. **Note that the y-axis starts from 50 span-F1.** STL indicates single-task learning and MTL indicates the multi-task model. Differences can be seen on the test set: JobSpanBERT performs best on **SKILLS**, JobBERT is best on **KNOWLEDGE**, and JobBERT achieves best in **COMBINED**. Exact numbers of the plots are in [Table 5.5 \(Subsection 5.9.7\)](#).

for each tag type (skill and knowledge). In the STL experiments we use one CRF for the given tag type.

We use the MACHAMP toolkit ([van der Goot et al., 2021b](#)) for our experiments. For each setup we do five runs (i.e., five random seeds).<sup>9</sup> For evaluation we use span-level precision, recall, and F1, where the F1 for the MTL setting is calculated as described in [Benikova et al. \(2014\)](#).

<sup>9</sup>For reproducibility, we refer to [Subsection 5.9.6](#).

## 5.6 Results

The results of the experiments are given in [Figure 5.3](#). We show the average performance of each model in F1 and respective standard deviation over the development and test split. Exact scores on each source split and other metric details are provided in [Subsection 5.9.7](#). As mentioned before, we experiment with the following settings: **SKILL**, we train and predict only on skills. **KNOWLEDGE**, train and only predict for knowledge. **COMBINED**, we merge the STL predictions of both skills and knowledge. We also train the models in an MTL setting, predicting both skills and knowledge simultaneously. We evaluate the MTL model on both **SKILL** and **KNOWLEDGE** separately, and also compare it against the aggregated STL predictions.

### 5.6.1 Performance on Development Set

In [Figure 5.3](#), we show the results on the development set in the upper plot. We observe similar performance between the domain-adapted STL models—JobBERT and JobSpanBERT—have similar span-F1 for **SKILL**:  $60.05_{\pm 0.70}$  vs.  $60.07_{\pm 0.70}$ . In contrast, for **KNOWLEDGE**,  $\text{BERT}_{\text{base}}$  and JobBERT are closest in predictive performance:  $60.44_{\pm 0.58}$  vs.  $60.66_{\pm 0.43}$ . In the **COMBINED** setting, JobBERT performs highest with a span-F1 of  $60.32_{\pm 0.39}$ . On average, JobBERT performs best over all three settings. Surprisingly, the models for both **SKILL** and **KNOWLEDGE** perform similarly (around 60 span-F1), despite the sources’ differences in properties and length [Figure 5.2](#). In addition, we find that MTL is not performing better than STL across sources. For exact numbers and source-level (i.e., **BIG**, **HOUSE**, **TECH**), we refer to [Subsection 5.9.7](#).

### 5.6.2 Performance on Test Set

We select the best performing models in the development set evaluation and apply it to the test set. Results are in [Figure 5.3](#) in the bottom plot. Since JobBERT and JobSpanBERT are performing similarly, we apply both to the

test set and  $BERT_{base}$ . We observe a deviation from the development set to the test set: JobSpanBERT  $60.07_{\pm 0.30} \rightarrow 56.64_{\pm 0.83}$  on SKILL, JobBERT  $60.66_{\pm 0.43} \rightarrow 63.88_{\pm 0.28}$  on KNOWLEDGE. For COMBINED, JobBERT performs slightly worse:  $60.32_{\pm 0.39} \rightarrow 59.73_{\pm 0.38}$ . Similar to the development set, we find that on all three methods of evaluation (i.e., SKILL, KNOWLEDGE, and COMBINED), STL still outperforms MTL. For SKILL and KNOWLEDGE, STL is almost stochastically dominant over MTL (i.e., significant), and for COMBINED there is stochastic dominance of STL over MTL, indicated in the next paragraph.

### 5.6.3 Significance

We compare all pairs of models based on five random seeds each using Almost Stochastic Order (ASO; Dror et al., 2019) tests with a confidence level of  $\alpha = 0.05$ . The ASO scores of the test set are indicated in Figure 5.4. We show that MTL-JobSpanBERT for SKILL shows almost stochastic dominance ( $\epsilon_{min} < 0.5$ ) over all other models. For KNOWLEDGE and COMBINED, We show that STL-JobBERT is stochastically dominant ( $\epsilon_{min} = 0.0$ ) over all the other models. For more details, we refer to Subsection 5.9.8 for ASO scores on the development set.

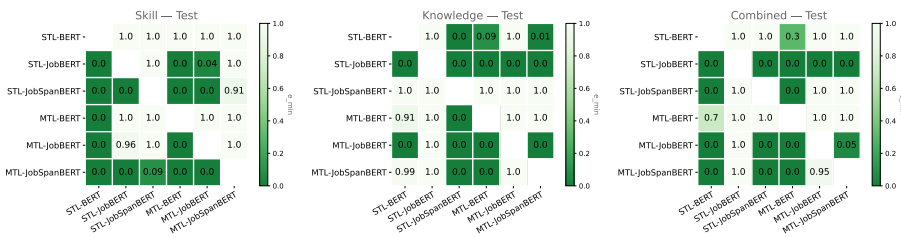


Figure 5.4: **Almost Stochastic Order Scores of the Test Set.** ASO scores expressed in  $\epsilon_{min}$ . The significance level  $\alpha = 0.05$  is adjusted accordingly by using the Bonferroni correction (Bonferroni, 1936). Read from row to column: E.g., in COMBINED STL-JobBERT (row) is stochastically dominant over  $STL-BERT_{base}$  (column) with  $\epsilon_{min}$  of 0.00.

## 5.7 Discussion

### 5.7.1 What Did Not Work

Additionally, we experiment whether representing the entire JP for extracting tokens yields better results than the experiments so far, which were sentence-by-sentence processing setups. To handle entire JPs and hence much longer sequences we use a pre-trained Longformer<sub>base</sub> (Beltagy et al., 2020) model. The document length we use in the experiments is 4096 tokens. Results of the Longformer on the test set are lower: For skills, JobSpanBERT against Longformer results in  $56.64_{\pm 0.83}$  vs.  $52.55_{\pm 2.39}$ . For KNOWLEDGE, JobBERT against Longformer shows  $63.88_{\pm 0.28}$  vs.  $57.26_{\pm 1.05}$ . Last, for COMBINED, JobBERT against Longformer results in  $59.73_{\pm 0.38}$  vs.  $55.05_{\pm 0.71}$ . This drop in performance is difficult to attribute to a concrete reason: e.g., the Longformer is trained on more varied sources than BERT, but not specifically for JPs, which may have contributed to this gap. Since the vanilla Longformer already performs worse than BERT<sub>base</sub> overall, we did not opt to apply domain-adaptive pre-training. Overall, we show that representing the full JP is not beneficial for SE, at least not in the Longformer setup tested here.

### 5.7.2 Continuous Pretraining helps SE

As previously mentioned, due to the domain specialization of the domain-adapted pre-trained BERT models, they predict more skills and frequently perform better in terms of precision, recall, and F1 as compared to their non-adaptive counterparts. This is especially encouraging as we confirm findings that continuous pre-training helps to adapt models to a specific domain (Alsentzer et al., 2019a; Lee et al., 2020a; Gururangan et al., 2020b; Nguyen et al., 2020a). However, there are exceptions. Particularly in Table 5.5 on TEST for KNOWLEDGE, BERT<sub>base</sub> comes closer in predictive performance to JobBERT (difference of 1.5 F1) than on SKILLS. Our intuition is that knowledge components are often already in the pre-training data (e.g., Wikipedia pages of certain competences like Python,

Java etc.) and therefore adaptive pre-training does not substantially boost performance.

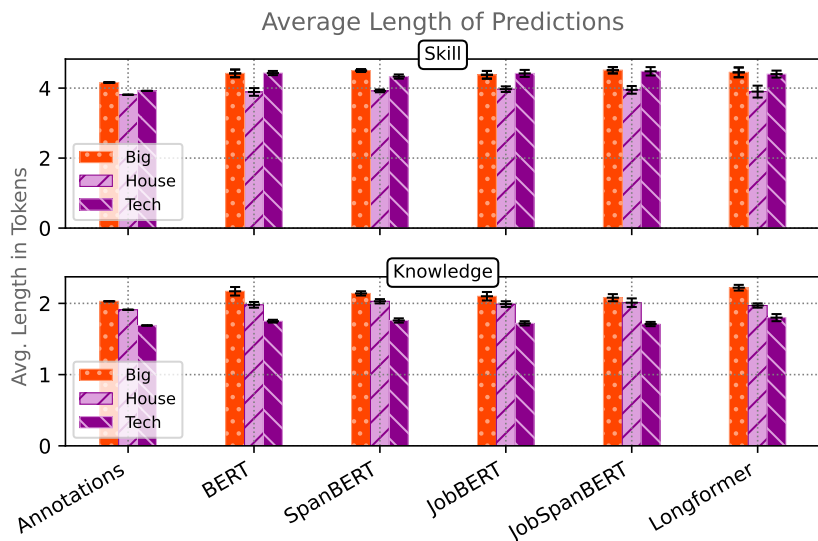
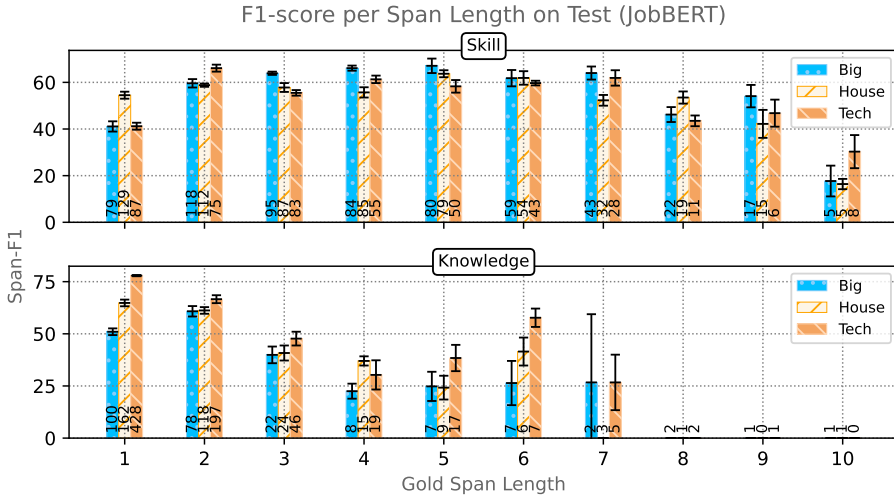


Figure 5.5: **Average Length of Predictions of Single Models.** We show the average length of the predictions versus the length of our annotated skills and knowledge components on the *test set* and the total number of predicted skills and knowledge tags in each respective split (#). There is a consistent trend over the three sources.

### 5.7.3 Difference in Length of Predictions

The main motivation of selecting models optimized for long spans was the length of the annotations (Figure 5.2). We investigate the average length of predictions of each model (Figure 5.5) to find out whether the models that are adapted to handle longer sequences truly predict longer spans. Interestingly, the average length of predicted skills are longer than the annotations over all three sources. There is a consistent trend among SKILL: BIG and TECH have similar length over predictions ( $>4$ ), while HOUSE is usually lower than length 3. For both BIG and TECH, JobSpanBERT predicts the longest skill spans (4.51 and 4.48 respectively). We suspect due to



**Figure 5.6: Average Span-F1 per Span Length.** We bucket the performance of JobBERT according to the length of the spans until 10 tokens and show the performance on each length, averaged over five random seeds. Indicated per bar is the support. The model performs best on medium-length skill spans (i.e., spans with token length of 4-5). For knowledge spans, on average, it performs best on short-length spans (i.e., spans with token length of 1-2).

the domain-adaptive pre-training on JPs, it improved the span prediction performance. In contrast, the Longformer predicts shorter spans. Note that the Longformer is not domain-adapted to JPs.

Regarding KNOWLEDGE, there is also a consistent trend: BIG has the overall longest prediction length while TECH has the lowest. The Longformer predicts the longest spans on average for BIG and TECH. Knowledge components are representative of a normal-length NER task and might not need a specialized model for long sequences. We show the exact numbers in [Table 5.7 \(Subsection 5.9.7\)](#) and the number of predicted SKILL and KNOWLEDGE: JobBERT and JobSpanBERT have higher recall than the other models.



### 5.7.4 Performance per Span Length

SKILLS are generally longer than KNOWLEDGE components in our dataset (Figure 5.2). The previous overall results on the test set (Figure 5.3) show that performance on SKILL is substantially lower than KNOWLEDGE. We therefore investigate whether this performance difference is attributed to the longer spans in SKILL. In Figure 5.6, we show the average performance of the best performing model (JobBERT) on the three sources (test set) based on the gold span length, until a length of 10.

In SKILL components (upper plot), we see much support for spans with length 1 and 2, which then lowers once the spans become longer. Spans with length of 1 shows low performance on BIG and TECH (around 40 span-F1), which influences the total span-F1. Short skills are usually soft skills, such as “passionate”, which can be used as a skill or not. This might confuse the model. In contrast, performance effectively stays similar (around 60 span-F1) for span length of 2 till 7 for all sources. Afterwards, it drops in performance. Thus, the weak performance on SKILL seem to be due to lower performance on the short spans.

For the KNOWLEDGE components (lower plot), they are generally shorter. We see that there is a gap in support between the sources, TECH has a larger number of gold labels compared to BIG and HOUSE. Unlike soft skills, KCs usually consist of proper nouns such as “Python”, “Java”, and so forth, which connects to the high performance on TECH (around 76 span-F1). Furthermore, support for spans longer than 2 drops considerably. In this case, if the model predicts a couple of instances correctly, it would substantially increase span-F1. Contrary to SKILL, high performance of KNOWLEDGE can be attributed to its strong performance on short spans.

## 5.8 Conclusion

We present a novel dataset for skill extraction on English job postings—SKILLSPAN—and domain-adapted BERT models—JobBERT and JobSpanBERT. We outline the dataset and annotation guidelines, created for hard

and soft skills annotation on the *span-level*. Our analysis shows that domain-adaptive pre-training helps to improve performance on the task for both skills and knowledge components. Our domain-adapted JobSpanBERT performs best on skills and JobBERT on knowledge. Both models achieve almost stochastic dominance over all other models for skills and knowledge extraction, whereas JobBERT in the STL setting achieves stochastic dominance over other models.

With the rapid emergence of new competences, our new approach of skill extraction has future potential, e.g., to enrich knowledge bases such as ESCO with unseen skills or knowledge components, and in general, contribute to providing insights into labor market dynamics. We hope our dataset encourages research into this emerging area of computational job market analysis.

## 5.9 Appendix

### 5.9.1 Data Statement SKILLSPAN

Following [Bender and Friedman \(2018\)](#), the following outlines the data statement for SKILLSPAN:

- A. **CURATION RATIONALE:** Collection of job postings in the English language for span-level sequence labeling, to study the impact of sequence labeling on the extraction of skill and knowledge components from job postings.
- B. **LANGUAGE VARIETY:** The non-canonical data was collected from the StackOverflow job posting platform, an in-house job posting collection from our national labor agency collaboration partner (The Danish Agency for Labor Market and Recruitment<sup>10</sup>), and web extracted job postings from a large job posting platform. US (en-US) and British (en-GB) English are involved.
- C. **SPEAKER DEMOGRAPHIC:** Gender, age, race-ethnicity, socioeconomic status are unknown.
- D. **ANNOTATOR DEMOGRAPHIC:** Three hired project participants (age range: 25–30), gender: one female and two males, white European and Asian (non-Hispanic). Native language: Danish, Dutch. Socioeconomic status: higher-education students. Female annotator is a professional annotator with a background in Linguistics and the two males with a background in Computer Science.
- E. **SPEECH SITUATION:** Standard American or British English used in job postings. Time frame of the data is between 2012–2021.
- F. **TEXT CHARACTERISTICS:** Sentences are from job postings posted on official job vacancy platforms.
- G. **RECORDING QUALITY:** N/A.
- H. **OTHER:** N/A.

---

<sup>10</sup><https://www.star.dk/en/>

- I. PROVENANCE APPENDIX: The job posting data in TECH is from Stack-overflow jobs, and is licensed under the CC BY-SA license. The job posting data from HOUSE is from our collaborators: The Danish Agency for Labour Market and Recruitment (STAR).

## 5.9.2 Annotation Guidelines

### 5.9.2.1 Span Specifications

**Legend:** Skill, Knowledge, “•” indicates an example sentence.

1. A skill starts with a **VERB**, otherwise (**ADJECTIVE**) + **NOUN**

1.1 Modal verbs are not tagged:

- Can [put personal touch on the menu]<sub>SKILL</sub> .
- Will [train new staff]<sub>SKILL</sub> .

## 2. Split up phrases with prepositions and/or conjunctions

2.1 **Unless** the conjunction coordinates two nouns functioning as one argument:

- [Coordinate parties and conferences]<sub>SKILL</sub> .

2.2 Do not tag skills with anaphoric pronouns, only tag preceding skill:

- [Prioritizing tasks]<sub>SKILL</sub> and identifying those that are most important.

2.3 Split nouns and adjectives that are coordinated if they do not have a verb attached:

- Be [inquisitive]<sub>SKILL</sub> and [proactive]<sub>SKILL</sub> .
- Prior in-house experience with [media]<sub>KNOWLEDGE</sub> , [publishing]<sub>KNOWLEDGE</sub> or [internet companies]<sub>KNOWLEDGE</sub> .

2.4 If there is a listing of skill tags and they lead up to different subtasks, we split them:

- [keep up the high level of quality in our team]<sub>SKILL</sub> through [reviews]<sub>SKILL</sub> , [pairing]<sub>SKILL</sub> and [mentoring]<sub>SKILL</sub> .

3. If there is relevant information appended after irrelevant information (e.g., info specific to a company) we try to make the skill as short as possible:

- [providing the best solution]<sub>SKILL</sub> ~~for Siemens Gamesa in a very~~ [structured]<sub>SKILL</sub> and [analytic]<sub>SKILL</sub> manner.

4. Note also the words skills and knowledge can be included in the span of the component if leaving it out makes it nonsensical:

- [personal skills]<sub>SKILL</sub> → just [personal] would make it nonsensical.

5. Parentheses after a skill tag are included if they elaborate the component before them or if they are an abbreviation of the component.

6. **Inclusion of adverbials in components.** Adverbials are included if it concerns the manner of doing something. All others are excluded:

- ~~like to~~ [solve technical challenges independently]<sub>SKILL</sub> .
- [communicates openly]<sub>SKILL</sub> .
- [striving for the best]<sub>SKILL</sub> ~~in all that they do.~~
- [Deliver first class customer service]<sub>SKILL</sub> ~~to our guests.~~
- [Making the right decisions]<sub>SKILL</sub> ~~early in the process.~~

7. **Attitudes as skills.** We annotate attitudes as a skill:

- a [can-do-approach]<sub>SKILL</sub> → we leave out articles from the attitude.

8. Attitudes are not tagged if they contain skill/knowledge components—then only the span of the skill is tagged.

- ~~like to~~ [solve technical challenges independently]<sub>SKILL</sub> .
- ~~Passion for~~ [automation]<sub>KNOWLEDGE</sub> .
- ~~enjoy~~ [working in a team]<sub>SKILL</sub> .

### 9. Miscellaneous:

9.1 Do not tag ironic skills (e.g., lazy).

9.2 Avoid nesting of skills, annotate it as one span.

9.3 We annotate all skills that are part of sections such as “requirements”, “good-to-haves”, “great-to-knows”, “optionals”, “after this  $x$  months of training you’ll be able to...”, “At the job you’re going to...”.

9.4 When there is a general standard that can be added to the skill, we add these:

- [Process payments according to the [...] standards]<sub>SKILL</sub> .

### 5.9.3 Knowledge Specifications

1. **Rule-of-thumb:** knowledge is something that one possesses, and cannot (usually) physically execute:

- [Python]<sub>KNOWLEDGE</sub> (programming language).
- [Business]<sub>KNOWLEDGE</sub> .
- [Relational Databases]<sub>KNOWLEDGE</sub> .

2. If there is a component between parentheses that belongs to the knowledge component, we add it:

- [(non-) relational databases]<sub>KNOWLEDGE</sub> .
- [Driver License (UK/EU)]<sub>KNOWLEDGE</sub> .

3. **Licenses and certifications:** We add the additional words “certificate”, “card”, “license”, et cetera. to the knowledge component.

4. If the knowledge component looks like a skill, but the preceding verb is vague and empty (e.g., *follow*, *use*, *comply with*, *work with*) → only tag the knowledge component:

- ~~Comply with~~ [Food Code of Practice]<sub>KNOWLEDGE</sub> .
- ~~Work with~~ [AWS infrastructure]<sub>KNOWLEDGE</sub> .

5. We annotate only specified knowledge components:

- [MongoDB]<sub>KNOWLEDGE</sub> or other [NoSQL database]<sub>KNOWLEDGE</sub> .
- [JEST]<sub>KNOWLEDGE</sub> or other test libraries. → “other test libraries” is under-specified.

6. Knowledge components can be nested in skill components.

- [Design, execution and analysis...]<sub>SKILL</sub>  
[of [phosphoproteomics]<sub>KNOWLEDGE</sub> experiments]<sub>SKILL</sub> .

7. If all components coordinate/share one knowledge tag, we annotate it as one:

- [application, data and infrastructure architecture]<sub>KNOWLEDGE</sub> . → The knowledge tags coordinate to “architecture”.
- [chemical/biochemical engineering]<sub>KNOWLEDGE</sub> .

8. If there is a listing of knowledge tags, we annotate all knowledge tags separately:

- [Bachelor Degree]<sub>KNOWLEDGE</sub> in [Mathematics]<sub>KNOWLEDGE</sub> ,  
[Computer Science]<sub>KNOWLEDGE</sub> , or [Engineering]<sub>KNOWLEDGE</sub> .



### 5.9.4 Other Specifications

1. **Rule-of-thumb:** If in doubt, annotate it as a skill.

2. We are preferring skills over knowledge components.

3. We prioritize skills over attitudes; if there is a skill within the attitude, only tag the skill:

- ~~Passionate around~~ [solving business problems]<sub>SKILL</sub> through [innovation & engineering practices]<sub>KNOWLEDGE</sub>.

4. Skill or knowledge components in the top headlines of the JP are not tagged (e.g., title of a JP). If it is a sub-headline or in the rest of the posting, tag it.

5. We try to keep the skill/knowledge components as short as possible (i.e., exclude information at the end if it makes it too specific for the job).

6. We do not include “fluff” and “triggers” (i.e., words that indicate a skill or knowledge component will follow: “~~advanced knowledge of~~ [...] <sub>KNOWLEDGE</sub>”) around the components, including degree. This goes for both before and after:

- ~~Working proficiency in~~ [developmental toolsets]<sub>KNOWLEDGE</sub>.
- ~~Knowledge of~~ [application data and architecture]<sub>KNOWLEDGE</sub> disciplines.
- [Manual handling]<sub>SKILL</sub> tasks.
- [CI/CD]<sub>KNOWLEDGE</sub> experience.
- ~~You master~~ [English]<sub>KNOWLEDGE</sub> on level C1.
- ~~Proficient in~~ [Python]<sub>KNOWLEDGE</sub> and [English]<sub>KNOWLEDGE</sub>.
- ~~Fluent in spoken and written~~ [English]<sub>KNOWLEDGE</sub>.

7. Pay attention to expressions such as “participation in...”, “contributing”, and “transfer (knowledge)”. These are usually not considered skills.

- ~~Contribute to the enjoyable and collaborative work environment.~~
- ~~Participation in the Department’s regular research activities.~~
- ~~Desire to be part of something meaningful and innovative.~~

8. Skills and Knowledge components that are found in not-so-straightforward places (e.g., project descriptions) are annotated as well, if they relate to the position.

9. In the pattern of “skill” followed by some elaboration, see if it can be annotated with a skill and a knowledge tag:

- [Ensure food storage and prep. areas are maintained]<sub>SKILL</sub> according to [Health & Safety and Audit standards]<sub>KNOWLEDGE</sub>.

10. Occupations and positions in companies/academia should be excluded.

11. If there’s a knowledge/skill component in the position, we exclude it as well.

- Experienced Java Engineer. → completely untagged.

<p>12. Only annotate the skills that are <u>related</u> to the position.</p> <p>12.1. This includes skills that are specific for the position as well (e.g., skills of a ruminants professor versus math professor).</p> <p>12.2 Also skills that the person for the position is expected to do in the future.</p> <p>12.3 This does <u>not</u> include skills, knowledge or attitudes describing only the company, the group you will join in the department, and so on. <u>Only annotate</u> if it is specified or implied that the employee should possess the skill as well.</p>
<p>13. We annotate industries and fields (that the employee will be working in) as knowledge components.</p>

### 5.9.5 Type of Skills Annotated

In both [Table 5.8](#) and [Table 5.9](#), we show the top-10 skill and knowledge components that have been annotated. We split the top-10 among the data splits (i.e., train, development, and test set), and also between source splits (i.e., BIG, HOUSE, TECH).

### 5.9.6 Reproducibility

We use the default hyperparameters in MACHAMP ([van der Goot et al., 2021b](#)) as shown in [Table 5.4](#). For more details we refer to their paper. For the five random seeds we use 3477689, 4213916, 6828303, 8749520, and 9364029. All experiments with MACHAMP were ran on an NVIDIA<sup>®</sup> TITAN X (Pascal) 12 GB GPU and an Intel<sup>®</sup> Xeon<sup>®</sup> Silver 4214 CPU.

### 5.9.7 Exact Number of Performance

In [Table 5.5](#), we show the exact numbers of the plot indicated in [Figure 5.3](#). In addition, we also show the results of each respective split.

Parameter	Value	Range
Optimizer	AdamW	
$\beta_1, \beta_2$	0.9, 0.99	
Dropout	0.2	0.1, 0.2, 0.3
Epochs	20	
Batch Size	32	
Learning Rate (LR)	1e-4	1e-3, 1e-4, 1e-5
LR scheduler	Slanted triangular	
Weight decay	0.01	
Decay factor	0.38	0.35, 0.38, 0.5
Cut fraction	0.2	0.1, 0.2, 0.3

Table 5.4: Hyperparameters of MACHAMP.

For the STL models, we observe differences in performances over the sources which is particularly pronounced for knowledge components: The TECH source is the easiest to process (and has most SKCs), while SKCs identification performance is the lowest for BIG. This might be due to the broad nature of this source.

In the exact results table (Table 5.5) we add a (†) next to the highest span-F1 if the model is truly stochastically dominant ( $\epsilon_{\min} = 0.0$ ) over *all* the other models. (\*) denotes that the best model achieved *almost stochastic dominance* ( $\epsilon_{\min} < 0.5$ ) over—at minimum—one other model (e.g., in TEST rows w.r.t COMBINED: MTL-JobBERT  $\succeq$  MTL-JobSpanBERT with  $\epsilon_{\min} = 0.06$ ) and stochastically dominant over the rest.

In Table 5.6, we report the precision and recall of the models, SKILL and KNOWLEDGE show the precision and recall of the STL models. MULTI shows the precision and recall of the MTL models.

Last, in Table 5.7, we show the exact numbers of the length of predictions Figure 5.5. We also add the number of predicted SKILL and KNOWLEDGE Overall, JobBERT and JobSpanBERT predict more skills in general than the other models. This is also the case for knowledge components. We hypothesize that this might be due to the BERT models now being more specialized towards the JP domain and recognizing more SKCs.

Evaluation →		SKILL		KNOWLEDGE		COMBINED	
Src.	↓ Model, Task →	STL	MTL	STL	MTL	STL (*2)	MTL
BIG	BERT <sub>base</sub>	59.55 $\pm$ 0.97	58.88 $\pm$ 1.14	50.68 $\pm$ 3.25	51.10 $\pm$ 1.67	57.46 $\pm$ 1.19	57.00 $\pm$ 0.91
	SpanBERT	59.78 $\pm$ 0.44	60.02 $\pm$ 2.15	50.65 $\pm$ 2.32	51.79 $\pm$ 2.12	57.71 $\pm$ 0.53	58.00 $\pm$ 2.07
	JobBERT	60.60 $\pm$ 0.81	59.76 $\pm$ 0.60	50.29 $\pm$ 1.86	47.59 $\pm$ 1.11	58.19 $\pm$ 0.49	56.75 $\pm$ 0.50
	JobSpanBERT	60.16 $\pm$ 0.61	59.44 $\pm$ 1.11	45.20 $\pm$ 2.76	47.69 $\pm$ 3.38	56.56 $\pm$ 0.49	56.58 $\pm$ 0.63
HOUSE	BERT <sub>base</sub>	56.83 $\pm$ 1.29	55.89 $\pm$ 1.90	55.00 $\pm$ 1.11	54.05 $\pm$ 1.00	56.17 $\pm$ 0.92	55.20 $\pm$ 1.35
	SpanBERT	57.54 $\pm$ 1.08	57.30 $\pm$ 0.84	52.01 $\pm$ 1.72	51.48 $\pm$ 1.01	55.55 $\pm$ 1.10	55.09 $\pm$ 0.74
	JobBERT	59.81 $\pm$ 1.17	59.97 $\pm$ 0.85	54.94 $\pm$ 1.15	54.23 $\pm$ 2.60	58.02 $\pm$ 0.93	57.80 $\pm$ 1.50
	JobSpanBERT	59.97 $\pm$ 1.03	59.62 $\pm$ 0.74	55.66 $\pm$ 1.51	53.10 $\pm$ 1.27	58.37 $\pm$ 1.07	57.14 $\pm$ 0.56
TECH	BERT <sub>base</sub>	59.05 $\pm$ 0.71	58.34 $\pm$ 0.75	64.08 $\pm$ 1.04	63.77 $\pm$ 1.18	62.10 $\pm$ 0.67	61.65 $\pm$ 0.62
	SpanBERT	58.39 $\pm$ 0.46	58.61 $\pm$ 1.14	62.68 $\pm$ 0.60	63.40 $\pm$ 0.93	61.02 $\pm$ 0.35	61.56 $\pm$ 0.81
	JobBERT	59.81 $\pm$ 0.75	59.36 $\pm$ 0.90	64.57 $\pm$ 0.42	63.15 $\pm$ 0.94	62.69 $\pm$ 0.40	61.67 $\pm$ 0.90
	JobSpanBERT	60.09 $\pm$ 1.43	59.48 $\pm$ 0.61	63.40 $\pm$ 1.51	63.23 $\pm$ 0.64	62.09 $\pm$ 0.85	61.80 $\pm$ 0.54
AVERAGE	BERT <sub>base</sub>	58.45 $\pm$ 0.68	57.67 $\pm$ 1.01	60.44 $\pm$ 0.58	59.98 $\pm$ 0.75	59.35 $\pm$ 0.46	58.72 $\pm$ 0.48
	SpanBERT	58.53 $\pm$ 0.33	58.60 $\pm$ 0.83	58.89 $\pm$ 0.49	59.21 $\pm$ 0.78	58.69 $\pm$ 0.36	58.88 $\pm$ 0.64
	JobBERT	60.05 $\pm$ 0.70	59.69 $\pm$ 0.62	<b>60.66<math>\pm</math>0.43</b> *	59.15 $\pm$ 1.07	<b>60.32<math>\pm</math>0.39</b> *	59.44 $\pm$ 0.81
	JobSpanBERT	<b>60.07<math>\pm</math>0.30</b> †	59.51 $\pm$ 0.68	59.47 $\pm$ 1.31	59.04 $\pm$ 0.65	59.79 $\pm$ 0.53	59.29 $\pm$ 0.43
TEST	BERT <sub>base</sub>	54.34 $\pm$ 0.74	54.20 $\pm$ 0.68	62.43 $\pm$ 0.41	61.66 $\pm$ 0.83	58.16 $\pm$ 0.47	57.73 $\pm$ 0.66
	JobBERT	56.11 $\pm$ 0.49	55.46 $\pm$ 0.75	<b>63.88<math>\pm</math>0.28</b> *	63.35 $\pm$ 0.30	<b>59.73<math>\pm</math>0.38</b> †	59.18 $\pm$ 0.37
	JobSpanBERT	<b>56.64<math>\pm</math>0.83</b> *	56.27 $\pm$ 0.55	61.06 $\pm$ 0.99	61.87 $\pm$ 0.55	58.72 $\pm$ 0.69	58.90 $\pm$ 0.48

Table 5.5: **Performance of Models.** We test the models on **skills**, **KNOWLEDGE**, and **COMBINED** (MTL). We report the span-F1 and standard deviation of runs on five random seeds on the *development set* (**AVERAGE**, in gray). Results on the *test set* are below in the **TEST** rows (in cyan). **STL** indicates single-task learning and **MTL** indicates the multi-task model. **Bold** numbers indicate best performing model in that experiment. A (†) means that it is stochastically dominant over *all* the other models. (\*) denotes *almost stochastic dominance* ( $\epsilon_{\min} < 0.5$ ) over—at minimum—one other model.

## 5.9.8 Significance Testing

Recently, the ASO test (Dror et al., 2019)<sup>11</sup> has been proposed to test statistical significance for deep neural networks over multiple runs. Generally, the ASO test determines whether a stochastic order (Reimers and Gurevych, 2018) exists between two models or algorithms based on their respective sets of evaluation scores. Given the single model scores over multiple random seeds of two algorithms  $\mathcal{A}$  and  $\mathcal{B}$ , the method computes a test-specific value ( $\epsilon_{\min}$ ) that indicates how far algorithm  $\mathcal{A}$  is from being

<sup>11</sup>Implementation of Dror et al. (2019) can be found at <https://github.com/Kaleidophon/deep-significance> (Ulmer, 2021)

Evaluation →		SKILL		KNOWLEDGE		MULTI	
Src.	↓ Model	Precision	Recall	Precision	Recall	Precision	Recall
BIG	BERT <sub>base</sub>	57.09 <sub>±1.70</sub>	62.27 <sub>±1.28</sub>	43.95 <sub>±4.17</sub>	60.00 <sub>±1.65</sub>	52.63 <sub>±1.32</sub>	62.19 <sub>±0.87</sub>
	SpanBERT	58.28 <sub>±0.59</sub>	61.36 <sub>±0.68</sub>	45.80 <sub>±2.89</sub>	56.82 <sub>±3.39</sub>	54.02 <sub>±1.81</sub>	62.63 <sub>±2.60</sub>
	JobBERT	57.90 <sub>±1.25</sub>	63.59 <sub>±0.99</sub>	43.45 <sub>±1.98</sub>	59.84 <sub>±3.44</sub>	51.13 <sub>±0.48</sub>	63.74 <sub>±0.79</sub>
	JobSpanBERT	58.39 <sub>±1.03</sub>	62.09 <sub>±1.85</sub>	38.55 <sub>±3.12</sub>	54.76 <sub>±3.18</sub>	52.22 <sub>±0.35</sub>	61.75 <sub>±1.22</sub>
HOUSE	BERT <sub>base</sub>	55.95 <sub>±2.46</sub>	57.79 <sub>±0.67</sub>	52.84 <sub>±0.65</sub>	57.42 <sub>±2.76</sub>	51.65 <sub>±1.11</sub>	59.28 <sub>±2.07</sub>
	SpanBERT	56.70 <sub>±1.59</sub>	58.44 <sub>±1.16</sub>	49.87 <sub>±2.57</sub>	54.49 <sub>±3.09</sub>	52.27 <sub>±0.64</sub>	58.25 <sub>±1.50</sub>
	JobBERT	58.16 <sub>±1.30</sub>	61.56 <sub>±1.53</sub>	51.18 <sub>±2.18</sub>	59.37 <sub>±1.34</sub>	53.72 <sub>±1.57</sub>	62.56 <sub>±1.47</sub>
	JobSpanBERT	59.04 <sub>±0.85</sub>	60.99 <sub>±2.58</sub>	51.36 <sub>±2.70</sub>	60.84 <sub>±1.19</sub>	53.91 <sub>±0.77</sub>	60.79 <sub>±0.54</sub>
TECH	BERT <sub>base</sub>	58.28 <sub>±1.30</sub>	59.89 <sub>±1.39</sub>	60.79 <sub>±1.89</sub>	67.79 <sub>±1.20</sub>	58.19 <sub>±1.12</sub>	65.55 <sub>±0.75</sub>
	SpanBERT	58.62 <sub>±0.32</sub>	58.16 <sub>±0.76</sub>	59.43 <sub>±1.21</sub>	66.35 <sub>±1.18</sub>	58.34 <sub>±0.97</sub>	65.17 <sub>±1.41</sub>
	JobBERT	58.81 <sub>±1.38</sub>	60.88 <sub>±1.51</sub>	61.38 <sub>±1.11</sub>	68.14 <sub>±1.36</sub>	57.69 <sub>±0.93</sub>	66.25 <sub>±0.90</sub>
	JobSpanBERT	59.86 <sub>±3.07</sub>	60.40 <sub>±0.68</sub>	59.78 <sub>±2.43</sub>	67.57 <sub>±1.97</sub>	58.26 <sub>±0.82</sub>	65.82 <sub>±0.91</sub>
AVERAGE	BERT <sub>base</sub>	57.11 <sub>±1.65</sub>	59.90 <sub>±0.95</sub>	56.86 <sub>±1.33</sub>	64.54 <sub>±1.31</sub>	55.02 <sub>±0.85</sub>	62.98 <sub>±0.93</sub>
	SpanBERT	57.85 <sub>±0.65</sub>	59.23 <sub>±0.52</sub>	55.65 <sub>±1.09</sub>	62.58 <sub>±1.56</sub>	55.61 <sub>±0.61</sub>	62.58 <sub>±1.25</sub>
	JobBERT	58.29 <sub>±1.08</sub>	61.94 <sub>±1.16</sub>	56.73 <sub>±1.41</sub>	65.22 <sub>±1.03</sub>	55.03 <sub>±0.84</sub>	64.62 <sub>±0.77</sub>
	JobSpanBERT	59.11 <sub>±1.59</sub>	61.12 <sub>±1.49</sub>	55.11 <sub>±2.41</sub>	64.66 <sub>±1.38</sub>	55.64 <sub>±0.56</sub>	63.46 <sub>±0.69</sub>
TEST	BERT <sub>base</sub>	56.02 <sub>±1.50</sub>	52.79 <sub>±1.18</sub>	59.09 <sub>±0.85</sub>	66.20 <sub>±1.69</sub>	55.82 <sub>±1.03</sub>	59.79 <sub>±0.87</sub>
	JobBERT	55.94 <sub>±1.19</sub>	56.29 <sub>±0.49</sub>	60.03 <sub>±1.13</sub>	68.30 <sub>±1.46</sub>	55.87 <sub>±0.29</sub>	62.89 <sub>±0.56</sub>
	JobSpanBERT	57.57 <sub>±1.24</sub>	55.77 <sub>±1.65</sub>	57.83 <sub>±1.03</sub>	64.71 <sub>±2.10</sub>	57.06 <sub>±0.74</sub>	60.89 <sub>±0.42</sub>

Table 5.6: **Precision and Recall of Models.** We test the models on *skills*, *knowledge*, and *multi-task* setting. We report the average precision, recall and standard deviation of runs on five random seeds on the *development set* (**AVERAGE**). Results on the *test set* are below in the **TEST** rows.

significantly better than algorithm  $\mathcal{B}$ . When distance  $\epsilon_{\min} = 0.0$ , one can claim that  $\mathcal{A}$  stochastically dominant over  $\mathcal{B}$  with a predefined significance level. When  $\epsilon_{\min} < 0.5$  one can say  $\mathcal{A} \succeq \mathcal{B}$ . On the contrary, when we have  $\epsilon_{\min} = 1.0$ , this means  $\mathcal{B} \succeq \mathcal{A}$ . For  $\epsilon_{\min} = 0.5$ , no order can be determined. We took 0.05 for the predefined significance level. In Figure 5.7, we show the ASO scores on the development set.

Source →	BIG		HOUSE		TECH	
↓ Model	SKILLS (#)	KNOWLEDGE (#)	SKILLS (#)	KNOWLEDGE (#)	SKILLS (#)	KNOWLEDGE (#)
ANNOTATIONS	4.16 (634)	2.03 (242)	3.81 (637)	1.91 (350)	3.92 (459)	1.69 (834)
BERT <sub>base</sub>	4.42 $\pm$ 0.11 (628)	2.17 $\pm$ 0.06 (307)	3.89 $\pm$ 0.11 (615)	1.98 $\pm$ 0.04 (461)	4.43 $\pm$ 0.06 (449)	1.75 $\pm$ 0.02 (885)
SpanBERT	4.50 $\pm$ 0.04 (621)	2.14 $\pm$ 0.03 (298)	3.92 $\pm$ 0.04 (597)	2.03 $\pm$ 0.03 (441)	4.33 $\pm$ 0.06 (444)	1.76 $\pm$ 0.03 (869)
JobBERT	4.38 $\pm$ 0.11 (670)	2.10 $\pm$ 0.06 (313)	3.97 $\pm$ 0.08 (650)	1.99 $\pm$ 0.04 (470)	4.42 $\pm$ 0.10 (479)	1.72 $\pm$ 0.03 (932)
JobSpanBERT	4.51 $\pm$ 0.09 (629)	2.08 $\pm$ 0.05 (313)	3.95 $\pm$ 0.11 (623)	2.01 $\pm$ 0.06 (452)	4.48 $\pm$ 0.12 (439)	1.71 $\pm$ 0.03 (875)
Longformer	4.45 $\pm$ 0.14 (653)	2.22 $\pm$ 0.04 (298)	3.90 $\pm$ 0.17 (639)	1.97 $\pm$ 0.03 (483)	4.40 $\pm$ 0.10 (472)	1.80 $\pm$ 0.05 (864)

**Table 5.7: Average Length of Predictions of Single Models.** We show the average length of the predictions versus the length of our annotated skills and knowledge components on the *test set* and the total number of predicted skills and knowledge tags in each respective split (#).

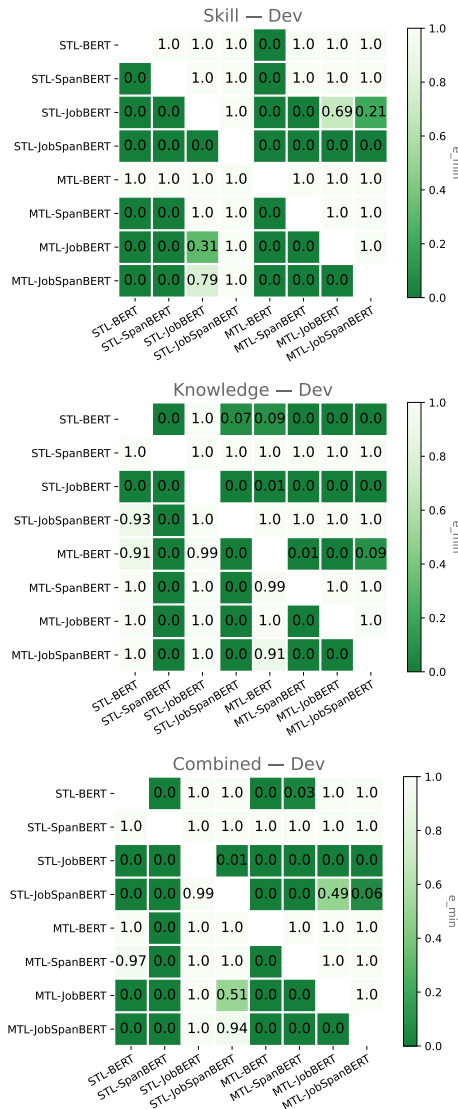


Figure 5.7: **Almost Stochastic Order Scores of the Development Set.** ASO scores expressed in  $\epsilon_{\min}$ . The significance level  $\alpha = 0.05$  is adjusted accordingly by using the Bonferroni correction (Bonferroni, 1936). Read from row to column: E.g., STL-JobBERT (row) is stochastically dominant over STL-BERT<sub>base</sub> (column) with  $\epsilon_{\min}$  of 0.00.



Skills			
Src.	Train	Test	
BIG	enthusiastic flexible team player friendly attention to detail communicator passionate confident flexible approach	ambitious proactive work independently attention to detail motivated reliable flexible willingness to learn self-motivated work as part of an established team	customer service communicator flexible attention to detail ambitious design and refine every touchpoint of the customer journey enable inclusion communicate effectively interpersonal skills proactive
	communication skills motivated structured structured proactive analytical communication self-driven team player teaching curious	structured teaching communication skills project management drive problem solving communication visit customers curious work independently	teaching research communication skills outgoing flexible energetic responsible enthusiastic team player communication
TECH	communication skills passionate apply your depth of knowledge and expertise partner continuously with your many stakeholders solving business problems through innovation and engineering practices work in large collaborative teams hands-on building an innovative culture team player develop	hands-on communication skills leadership passionate open-minded code reviews independent software development pioneer new approaches analytical skills	solving business problems apply your depth of knowledge and expertise partner continuously with your many stakeholders achieve organizational goals building an innovative culture stay focused on common goals work in large collaborative teams design development communicate

Table 5.8: Most Frequent Skills in the Data. Top-10 skill components in our data in terms of frequency.

	Knowledge		
Src.	Train	Development	Test
BIG	english	full uk driving licence	strategic planning
	driving license	sap energy assessments	english
	excel	right to work in the uk	cscs card
	cscs card	sen	pms
	maths	acca/aca	reservation systems
	ppc	professional kitchen	keynote
	service design	cra calculations	illustrator
	uk/emea policies	email marketing	aba
	bachelor's degree	qualitative and quantitative social research methods	sen
	computer science	care setting	full driving license
HOUSE	english	english	english
	engineering	supply chain	danish
	computer science	project management	business
	product management	powders	java
	python	machine learning	marketing
	finance	phd degree	plm
	project management	muscle models with learning and adaptation	production
	agile	walking robots	supply chain
	danish	model rules	economics
	javascript	capacity development	excel
TECH	javascript	java	java
	python	javascript	python
	java	aws	.net
	agile	docker	financial services
	financial services	node.js	c#
	node.js	typescript	javascript
	english	react	cloud
	kubernetes	linux	english
	cloud	amazon-web-services	reactjs
	docker	devops	automation

Table 5.9: **Most Frequent Knowledge in the Data.** Top-10 knowledge components in our data in terms of frequency.

# Chapter 6

## Kompetencer: Fine-grained Skill Classification in Danish Job Postings via Distant Supervision and Transfer Learning

The work presented in this chapter is based on a paper that has been published as: Mike Zhang, Kristian Nørgaard Jensen, and Barbara Plank. Kompetencer: Fine-grained skill classification in Danish job postings via distant supervision and transfer learning. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 436–447, Marseille, France, June 2022b. European Language Resources Association. URL <https://aclanthology.org/2022.lrec-1.46>.

## Abstract

Skill Classification (SC) is the task of classifying job competences from job postings. This work is the first in SC applied to Danish job vacancy data. We release the first Danish job posting dataset: KOMPETENCER (*en*: competences), annotated for nested spans of competences. To improve upon coarse-grained annotations, we make use of The European Skills, Competences, Qualifications and Occupations (ESCO; [le Vrang et al., 2014](#)) taxonomy API to obtain fine-grained labels via distant supervision. We study two setups: The zero-shot and few-shot classification setting. We fine-tune English-based models and RemBERT ([Chung et al., 2020](#)) and compare them to in-language Danish models. Our results show RemBERT significantly outperforms all other models in both the zero-shot and the few-shot setting.

## 6.1 Introduction

Job Posting data (JPs) is emerging on a variety of platforms in big quantities, and can provide insights on labor market skill set demands and aid job matching ([Balog et al., 2012](#)). *Skill Classification* (SC) is to classify competences (i.e., hard and soft skills) necessary for any occupation from unstructured text or JPs.

Several works focus on Skill Identification ([Jia et al., 2018](#); [Sayfullina et al., 2018](#); [Tamburri et al., 2020](#)). This is to classify whether a skill occurs in a sentence or job description. However, continuing the pipeline, there is little work in further categorizing the identified skills by leveraging taxonomies such as ESCO. Another limitation is the scope of language, where all previous work focus on English job postings. This hinders in particular local job seekers from finding an occupation suitable to their specific skills within their community via online job platforms.

In this work, we look into the Danish labor market. We introduce KOMPETENCER, a novel Danish job posting dataset annotated on the *span-level* for nested *Skill* and *Knowledge* Components (SKCs) in job postings. We

do not directly annotate for the fine-grained taxonomy codes from e.g., ESCO, but rather annotate more generic spans of SKCs (Figure 6.2), and then exploit the ESCO API to bootstrap fine-grained SKCs via distant supervision (Mintz et al., 2009) and create “silver” data for skill classification. Our proposed distant supervision pipeline is denoted in Figure 6.1.

Recently, Natural Language Processing has seen a surge of several transfer learning methods and architecture which help improve state-of-the-art significantly on several tasks (Peters et al., 2018; Howard and Ruder, 2018; Radford et al., 2018; Devlin et al., 2019b). In this work, we explore the benefits of zero-shot cross-lingual transfer learning with English BERT<sub>base</sub> (Devlin et al., 2019b) and a BERT<sub>base</sub> that we continuously pretrain (Han and Eisenstein, 2019; Gururangan et al., 2020b) on 3.2M English JP sentences and test on Danish and compare it to in-language models: Danish BERT and our domain-adapted Danish BERT model on 24.5M Danish JP sentences. We analyze the zero-shot transfer of English to Danish SC. Last, we experiment with few-shot training: We fine-tune a multilingual model (Chung et al., 2020) on English JPs with a few Danish JPs and show how zero-shot transfer compares to training on a small amount of in-language data.

**Contributions.** ① We release KOMPETENCER,<sup>1</sup> the first Danish Skill Classification dataset with distantly supervised fine-grained labels using the ESCO taxonomy. ② We furthermore present experiments and analysis with in-language Danish models vs. a zero-shot cross-lingual transfer from English to Danish with domain-adapted BERT models. ③ We target a few-shot learning setting with a multilingual model trained on both English and a few Danish JPs.

## Distant Supervision for Skill Classification

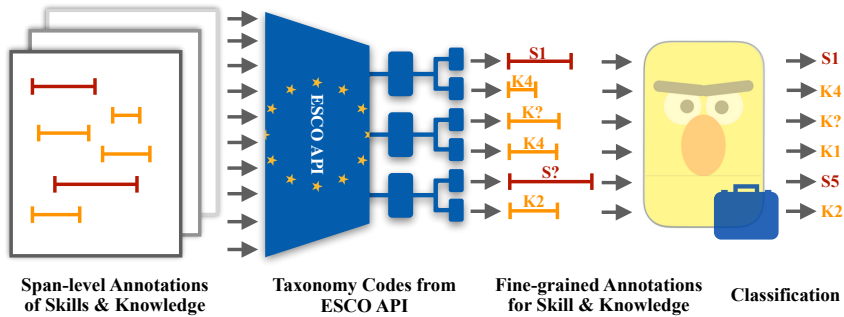


Figure 6.1: **Pipeline for Fine-grained Danish Skill Classification.** We propose a distant supervision pipeline, where we have identified spans of skills and knowledge. We query the ESCO API and fine-tune a model on the distantly supervised labels.

## 6.2 KOMPETENCER Dataset

### 6.2.1 Skill & Knowledge Definition

There is an abundance of competences and there have been large efforts to categorize them. The European Skills, Competences, Qualifications and Occupations (ESCO; le Vrang et al., 2014) taxonomy is the standard terminology linking skills, competences and qualifications to occupations. The ESCO taxonomy mentions three categories of competences: *Knowledge*, *skill*, and *attitudes*. ESCO defines knowledge as follows:

“Knowledge means the outcome of the assimilation of information through learning. Knowledge is the body of facts, principles, theories and practices that is related to a field of work or study.”<sup>2</sup>

For example, a person can acquire the Python programming language through learning. This is denoted as a *knowledge* component and can be

<sup>1</sup><https://github.com/jjzha/kompetencer>

<sup>2</sup>[ec.europa.eu/esco/portal/escopedia/Knowledge](http://ec.europa.eu/esco/portal/escopedia/Knowledge)

↓ Statistics, Language →	ENGLISH (EN)	DANISH (DA)
# Posts	391	60
# Sentences	14,538	1,479
# Tokens	232,220	20,369
# Skill Spans	6,576	665
# Knowledge Spans	6,053	255
$\bar{x}$ Skill Span	3.97	3.71
$\bar{x}$ Knowledge Span	1.80	1.73
$\tilde{x}$ Skill Span	4	3
$\tilde{x}$ Knowledge Span	1	1
Skill [90%]	[1, 9]	[1, 9]
Knowledge [90%]	[1, 5]	[1, 4]
Silver fine-grained labels	✓	✗
Gold fine-grained labels	✗	✓

Table 6.1: **Statistics of Annotated Dataset.** We report the total number of JPs across languages and their respective number of sentences, tokens, and SKCs. Below, we show the mean length of SKCs ( $\bar{x}$ ), median length of SKCs ( $\tilde{x}$ ), and the 90th percentile of length [90%] starting from length 1. We also indicate the type of labels in both sets (silver or gold labels). The EN set is larger than the DA split.

considered generally a *hard skill*. However, one also needs to be able to apply the knowledge component to a certain task. This is known as a *skill* component. ESCO formulates it as:

“Skill means the ability to apply knowledge and use know-how to complete tasks and solve problems.”<sup>3</sup>

In ESCO, the *soft skills* are referred to as *attitudes*. ESCO considers attitudes as skill components:

“The ability to use knowledge, skills and personal, social and/or methodological abilities, in work or study situations and professional and personal development.”<sup>4</sup>

<sup>3</sup>[ec.europa.eu/esco/portal/escopedia/Skill](http://ec.europa.eu/esco/portal/escopedia/Skill)

<sup>4</sup>[data.europa.eu/esco/skill/A](http://data.europa.eu/esco/skill/A)

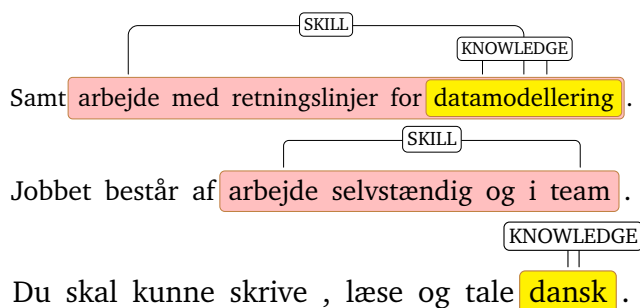


Figure 6.2: **Examples of Skills and Knowledge Components.** Annotated samples of passages in varying Danish job postings. SKCs can be nested as shown in the first example.

To sum up, hard skills are usually referred to as *knowledge* components, and applying these hard skills to something is considered a *skill*. Then, soft skills are referred to as *attitudes*, these are part of skill components. There has been no work, to the best of our knowledge, in annotating skill and knowledge components in JPs.

## 6.2.2 Dataset Statistics

Both the English and Danish data comes from a large job platform with various types of JPs.<sup>5</sup> The English JPs are from Zhang et al. (2022a). In Table 6.1, we show the statistics of both the annotated English and Danish data split. We note that the number of English JPs is larger than the Danish split. For Danish, there are fewer knowledge spans proportional to English. Apart from this, both the English and Danish JPs follow a similar trend in terms of statistics. The mean length of skills and knowledge ( $\bar{x}$ ) is slightly shorter for Danish, 3.97 vs. 3.71 and 1.80 vs. 1.73 respectively. The median length of skills ( $\tilde{x}$ ) is one token shorter for Danish. However, we note again that the length of skills can vary substantially, ranging from 1–9 for both languages. Then, for knowledge components this ranges from 1–5 and 1–4 for English and Danish respectively. The similarity in statistics

<sup>5</sup>We release the annotated spans in <https://github.com/jjzha/kompetencer/tree/master/data>



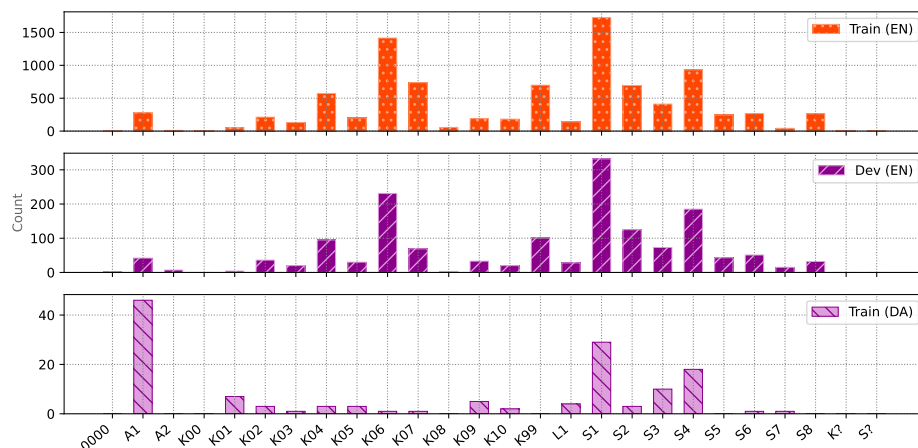


Figure 6.3: **Label Distribution of Distantly Supervised Labels.** In the top and middle barplot we show the fine-grained label distribution of the English training and development split respectively. The splits follow a similar distribution. For the Danish training split on the bottom, there is a large increase of A1 labels, which indicate more *attitude*-like skills. All splits have a larger fraction of the label S1, which encapsulates communicative skills. Explanations of labels are given in Table 6.3, Table 6.4, and Table 6.5 (Subsection 6.7.3, Appendix).

shows the consistency of annotations, which we elaborate on in the next section.

Figure 6.2 shows some examples of the annotated SKCs. “Samt arbejde med retningslinjer for datamodellering” (*en*: “As well as working with guidelines for data modeling”), shows a nesting example: “datamodellering” shows a knowledge component (i.e., something that one can learn), and the skill is to apply it. “Jobbet består af arbejde selvstændig og i team” (*en*: The job consists of working independently and in a team) indicates an *attitude* as “working independently or in a team” is a social ability. We furthermore consider languages a knowledge component, as one can acquire the language through schooling. Overall, the classification of the spans could be a short sentence (i.e.,  $\leq 9$  tokens) or single token classification.

### 6.2.3 Annotations

**Skill Identification Annotations** We annotate with the annotation guidelines denoted in Zhang et al. (2022a) used on the English data split to identify the SKCs in a JP. There are around 57.5K tokens (approximately 4.6K sentences, in 101 job posts) that was used to calculate agreement on. The annotations were compared using Cohen’s  $\kappa$  (Fleiss and Cohen, 1973) between pairs of annotators, and Fleiss’  $\kappa$  (Fleiss, 1971), which generalizes Cohen’s  $\kappa$  to more than two concurrent annotations. We consider two levels of  $\kappa$  calculations: **TOKEN** is calculated on the token level, comparing the agreement of annotators on each token (including non-entities) in the annotated dataset. **SPAN** refers to the agreement between annotators on the exact span match over the surface string, regardless of the type of named entity, i.e., we only check the position of tag without regarding the type of the named entity. The observed agreement scores over the three annotators is between 0.70–0.75 Fleiss’  $\kappa$  for both levels of calculation which is considered a *substantial agreement* (Landis and Koch, 1977). Then, for the Danish data split, we use the same guidelines as for English. Here, we consider one annotator that annotates for the SKCs.

**Fine-grained Annotations** Currently, our proposed dataset consists of identified SKCs. To obtain fine-grained labels of each span, we explore distant supervision using the ESCO API, where the setup is broadly depicted in Figure 6.1. The annotated spans are queried to the API, then via Algorithm 2, we determine whether the obtained SKC is “relevant” or not via Levenshtein distance matching (Levenshtein, 1966). In addition, we determine the quality of the distant supervised labels by human evaluation. We manually check each of the annotated spans to its obtained label from the ESCO API. After checking a subset 2,622 English labels — without correcting — and its distantly supervised labels, we obtain 41.3% accuracy on the correctness of the distantly supervised labels. We note that across all 9,473 labels in the original English training and development data (details of train/dev/test splits in Subsection 6.3.3), a total of 7.4% is

---

**Algorithm 2:** Getting the best match for a skill in the ESCO API using Levenshtein distance
 

---

```

1  $X \leftarrow$  Top-100 query results from ESCO;
2  $X \leftarrow \{x \mid \text{typeof}(x) = \text{Type}\}$ ;
3  $d \leftarrow \infty$ ;
4  $r \leftarrow \text{None}$ ;
5 for  $x \in X$  do
6     |  $\hspace{12em}$  // Find Skill in the ESCO API
7      $D \leftarrow \text{levenshtein}(x, \text{Skill})$ ;
8     if  $D = 0$  then
9         | return  $x$ ;  $\hspace{2em}$  // Perfect match
10    end
11    if  $D < d$  then
12        |  $r \leftarrow x$ ;
13        |  $d \leftarrow D$ ;
14    end
15 end
16 return  $r$ ;  $\hspace{2em}$  // Best match based on Levenshtein distance

```

---

unidentified by the ESCO database, and is thus labeled by K99 from ESCO in the resulting train and development data here. For the Danish data, we obtain 70.4% accuracy on the training set and 20.2% is missing, albeit the Danish training set only contains 138 SKCs. For the Danish test set, we correct the distantly supervised labels to create a gold test set. Here, 14.1% was initially correct and 23.5% missed a label. In [Figure 6.3](#), we show the distantly supervised fine-grained label distribution of the English training and development set split, and the Danish training split. The following labels: 0000, K?, and S? are artifacts of querying the ESCO API (i.e., unidentified skills). We did not employ any post-processing and left them as is. We presumed they would not influence the model significantly as their numbers are low.

### Experimental Pipeline Skill Classification

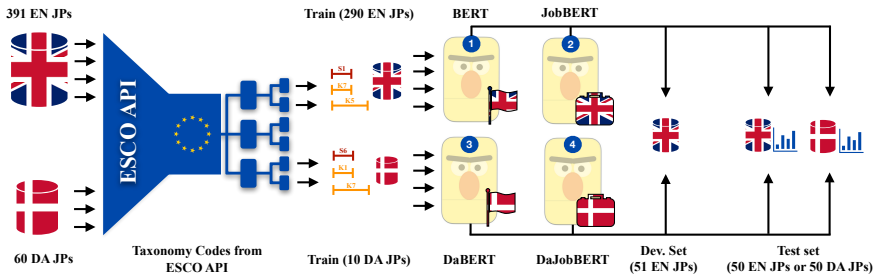


Figure 6.4: **Experimental Pipeline for Fine-grained Danish Skill Classification.** Read from left to right, we start with each respective dataset for English (EN) and Danish (DA). We obtain the labels from the ESCO API and train for each language split two models: For EN, these are (1) BERT<sub>base</sub> and (2) JobBERT. For DA, these are (3) DaBERT and (4) DaJobBERT. The Danish data is split into 10/50 train/test and the English data in to 290/51/50 train/dev/test JPs. The Danish models are fine-tuned on the Danish train set, and use *no* in-language development set (i.e., English dev.). In the end all models are applied to the Danish and English test set separately.

## 6.3 Methodology

In the current setting, we have annotated spans of SKCs. We extract the spans from the JPs and query the ESCO API to obtain silver labels. We formulate this task as a text classification problem. We consider a set of JPs  $\mathcal{D}$ , where  $d \in \mathcal{D}$  is a set of extracted spans (*and not full sentences*) with the  $i^{\text{th}}$  span  $\mathcal{X}_d^i = \{x_1, x_2, \dots, x_T\}$  and a target class  $c \in \mathcal{C}$ , where  $\mathcal{C} = \{S^*, K^*\}$ . The labels  $S^*$  and  $K^*$  depend on the distantly supervised ESCO taxonomy code (e.g., S4: Management Skills,<sup>6</sup> K2: Arts and Humanities<sup>7</sup>). The goal of this task is then to use  $\mathcal{D}$  to train an algorithm  $h : \mathcal{X} \mapsto \mathcal{C}$  to accurately predict skill tags by assigning an output label  $c$  for input  $\mathcal{X}_d^i$ .

<sup>6</sup><http://data.europa.eu/esco/skill/S4>

<sup>7</sup><http://data.europa.eu/esco/isced-f/02>

### 6.3.1 Encoders

As baseline for Danish SC, we consider a Danish BERT (**DaBERT**) encoder.<sup>8</sup> Following Gururangan et al. (2020b), we continuously pretrain DaBERT on 24.5M Danish JP sentences for *one* epoch, we name this **DaJobBERT**.<sup>9</sup> To test zero-shot performance from English to Danish for SC, we use **BERT**<sub>base</sub> (Devlin et al., 2019b) and a domain-adapted BERT<sub>base</sub> model on 3.2M JP sentences, namely **JobBERT** (Zhang et al., 2022a). We assume that domain-adapted models like JobBERT and DaJobBERT would improve SC as the “domain” is the same.

### 6.3.2 Multilingual Encoder

We explore whether using a multilingual encoder would benefit the classification of skills for Danish in a low-resource setting. For the experiments we use **RemBERT** (Chung et al., 2020), it has recently shown to outperform mBERT (Devlin et al., 2019b) on several tasks. All models are using a final Softmax layer for the classification of spans.

### 6.3.3 Experimental Setup

Our detailed experimental setup is shown in Figure 6.4. We start with 391 English and 60 Danish job postings (Table 6.1) annotated with spans of SKCs. The spans are then queried to the ESCO API (Figure 6.1). We split the English data into 290 train (9,472 SKCs), 51 dev (1,577 SKCs), and 50 JPs for test (1,578 SKCs), and for the Danish data we split this into 10 JPs (138 SKCs) for training and 50 JPs for test (782 SKCs). For the label distribution we refer back to Figure 6.3 (excl. test).

We fine-tune BERT<sub>base</sub> and JobBERT on the spans in 290 English JPs. Next, we fine-tune DaBERT and DaJobBERT on the 10 Danish JPs. For RemBERT, we fine-tune in three ways: Only on English, only on Danish, and on both English and Danish together. For all setups, we choose the

---

<sup>8</sup><https://huggingface.co/Maltehb/danish-bert-botxo>

<sup>9</sup><https://huggingface.co/jjzha/dajobbert-base-cased>

model with the best score on the English dev. set. As pointed out by [Artetxe et al. \(2020\)](#): Pure unsupervised cross-lingual transfer should not use any cross-lingual signal by definition. As our attention is on Danish, we do not use any Danish labeled training data *nor* dev. data in the zero-shot setting. All models in the end will be tested on the held-out 50 English and Danish JPs separately.<sup>10</sup> In summary, we have three setups: (1) Fine-tuned on English JPs only (BERT, JobBERT, RemBERT), (2) fine-tuned on Danish JPs only (DaBERT, DaJobBERT, RemBERT), and (3) fine-tuned on both English and Danish JPs (RemBERT). We consider (1) a zero-shot setting, while (2) and (3) do have access to some Danish training data, hence this is a few-shot setting. Throughout the experiments, we use the MACHAMP (v0.3) toolkit ([van der Goot et al., 2021b](#)) for classification. All reported results are the average over five runs with different random seeds on weighted macro-F1.

## 6.4 Analysis of Results

We show the experimental results in [Figure 6.5](#). Plotted is the weighted macro-F1 of all three setups with seven models and their corresponding standard deviation on the English development set, English test set, and the Danish test set. All models left of the black vertical line are the zero-shot setup, applied to Danish. On the right, these models are in the few-shot setting, this is due to the model having access to some target language training data (DA).

### 6.4.1 Performance Zero-shot Setting

For the models trained on English only (BERT, JobBERT, and RemBERT (EN)) when applied to the English development set, all three models perform similarly. They achieve around 0.63–0.64 weighted macro-F1 with little standard deviation:  $\text{BERT}_{\text{base}} 0.628_{\pm 0.004}$ , JobBERT  $0.628_{\pm 0.006}$ , and

---

<sup>10</sup>The English test set contains silver labels (distantly supervised), while the Danish test set is human corrected (gold).

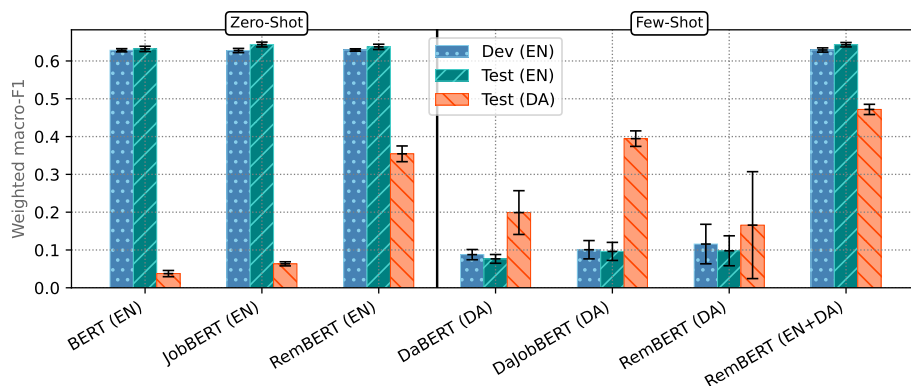


Figure 6.5: **Performance of Models on English and Danish.** We test seven setups on several splits of data: English development (**DEV (EN)**), English test set (**TEST (EN)**), and Danish test set (**TEST (DA)**). Reported is the weighted macro-F1. The whiskers indicate each respective standard deviation of runs on five random seeds. Left side of the black vertical line indicates a full zero-shot setting on **TEST (DA)**, on the right shows the few-shot setting on the same test set. With respect to the models, language abbreviation in brackets (e.g., **BERT (EN)**) indicates what it has been fine-tuned on. Exact numbers including significance testing are noted in [Table 6.6 \(Subsection 6.7.4, Appendix\)](#).

RemBERT (EN)  $0.629_{\pm 0.003}$  weighted macro-F1. Similarly for the English test set:  $\text{BERT}_{\text{base}}$   $0.632_{\pm 0.007}$ , JobBERT  $0.644_{\pm 0.006}$ , and RemBERT (EN)  $0.637_{\pm 0.007}$  weighted macro-F1, where JobBERT significantly outperforms all other models (details in [Subsection 6.7.4](#)).

It is a tacit that the English-based models perform better than the baseline (DaBERT) on English, both dev. and test. Conversely, the English-based models perform poorly on the Danish test set:  $\text{BERT}_{\text{base}}$   $0.038_{\pm 0.008}$  and JobBERT  $0.063_{\pm 0.005}$  weighted macro-F1. However, given a multilingual encoder (RemBERT) only trained on English, gives a significant gain in zero-shot performance ( $0.354_{\pm 0.021}$ ) with little standard deviation and significantly outperforms the other zero-shot setting models including the target-language baseline (DaBERT). We strongly suspect this is due to

Danish being included in the pretraining data of RemBERT.

### 6.4.2 Performance Few-shot Setting

Apart from RemBERT (EN+DA) having access to English data, all other models fine-tuned on Danish perform poorly on English dev. and test. The performance of RemBERT (DA) is slightly better than the best performing Danish-only model DaJobBERT ( $0.098_{\pm 0.040}$  vs.  $0.096_{\pm 0.024}$  weighted macro-F1 on English test), where our intuition again goes to the pretraining data.

For DA test, DaBERT is a strong baseline, achieving  $0.199_{\pm 0.058}$  weighted macro-F1 with little Danish training data. RemBERT (DA) did not result in significant gains having pretrained on multiple languages and another intuition could be that this is a result of negative transfer (Rosenstein et al., 2005). Then, DaJobBERT performs better than DaBERT on the Danish test set:  $0.395_{\pm 0.021}$  weighted macro-F1. Note that we conducted domain adaptive pretraining from the DaBERT checkpoint on 24.5M Danish JP sentences for one epoch with the Masked Language Modeling objective. This shows that in-language *and* in-domain pretraining is beneficial for this specific task of SC.

### 6.4.3 Combining Training Data

Last, giving RemBERT all training data (English and Danish) results in substantial improvement over all other models in the zero-shot and few-shot setting alike:  $0.472_{\pm 0.014}$ , which significantly outperforms all other models on Danish test. Henceforth, it is helpful to have a bit of target-language training data for higher resulting performance.

### 6.4.4 Is Domain Adaptive Pretraining Worth It?

In light of the results, domain adaptive pretraining shows its benefit for both English and Danish fine-tuning. Specifically for Danish, from the baseline (DaBERT), the improvement is close to 0.2 weighted macro-



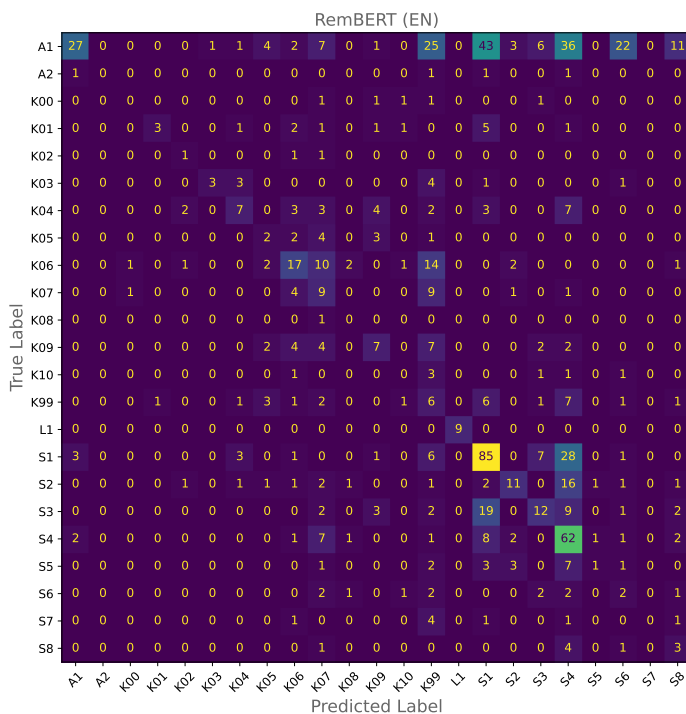


Figure 6.6: **Confusion Matrix of RemBERT (EN)**. We show the confusion matrix of the zero-shot setting with RemBERT (EN). On the diagonal are the correctly predicted labels. Most of the “confusion” is with respect to the labels that encompass the larger fraction of the test set: A1: Attitudes and S1: Communication, collaboration and creativity.

F1 with DaJobBERT. The domain adaptive pretraining took  $\sim 35$  hours, using 4 GPUs, to pass once over the unlabeled data (24.5M Danish JP sentences). The largest gain is obtained with combining both English and Danish training data: The improvement is around 0.27 weighted macro-F1. However, the 391 EN and 60 DA JPs took around two months of non-stop annotating. In short, there is a trade-off between continuous pretraining on unlabeled text and annotating: (1) Domain adaptive pretraining gives short-term gains with little costs, but there needs to be enough unlabeled data in the right domain. (2) Annotating extra data results in larger gains long-term, but there is more costs involved.

### 6.4.5 Analysis of Predictions

In [Figure 6.6](#), we show the confusion matrix of the best performing zero-shot model on the test set of the best run and investigate what the model does not predict correctly. In the matrix, the model mostly confuses the label A1, which relates to *attitudes* and gets predicted as S1: Communication, collaboration and creativity. There could be some overlap between these labels as for example the skill “effektiv” (*en*: efficient/effective). This is officially labeled as an attitude by ESCO, but a grey area is that “effective” could relate to “creativity”.

There is also a small cluster of confusion from S1-4. These are rather distinct classes of skills. For example, S4 means management skills. A specific example is “fagligt velfunderet” (*en*: professionally sound), this could be an attitude. This is all hard to determine since there is no context around the skill. Overall, there is some confusion between the skills when taken out of context. We leave the exploration of fine-grained skill classification *with context* for future work.

### 6.4.6 Qualitative Analysis Distant Supervision

We analyze the label selection method and the missing labels in the English dataset as mentioned in [Line 16](#). We find that the missing labels in the English data is predominantly coming from technical skills. We found that the missing spans are mostly knowledge components in the form of technologies used today by developers, such as ReactJS, Django, AWS, etc. This lack of coverage could either be due to specificity or the ever-growing list of technologies. In ESCO, there are several technologies that are listed (e.g., NoSQL, Drupal, WordPress to name a few), but there are also a lot missing (e.g., TensorFlow, Data Science, etc.).

## 6.5 Related Work

Many works focus on the identification of skills in job descriptions, i.e., whether a sentence contains a skill or not ([Sayfullina et al., 2018](#); [Tamburri](#)

et al., 2020) or what the necessary skills are inferred from an entire job posting (Bhola et al., 2020). We instead identified the SKCs manually in the job descriptions on the sentence-level, as this gives us the highest quality of identified SKCs. Furthermore, there are several works in fine-grained SC (i.e., categorize the skills), but mostly focus on English job descriptions. A straightforward approach is to do exact matching with a predefined list of skills (Malherbe and Aufaure, 2016; Papoutsoglou et al., 2017; Sibarani et al., 2017) or do a frequency analysis of skills, cluster them by hand and attach a more general category to them e.g., Gardiner et al. (2018).

Some works have used the ESCO taxonomy directly (Boselli et al., 2018; Giabelli et al., 2020). For example, Boselli et al. (2018) classified both titles and descriptions for its most suitable ISCO (Elias, 1997) code (what ESCO is partially based on). However, they only gave one label to each data point (i.e., full job posting), which is unrealistic as most occupations require multiple competences.

Overall, to the best of our knowledge, there seems to be little to no work in directly classifying the identified SKC to a specific ESCO label. In addition, this work is the first of its kind for Danish JPs.

## 6.6 Conclusion

We present a novel skill classification dataset for competences in Danish: KOMPETENCER.<sup>11</sup> In addition, we transform the coarse-grained human annotated spans to more fine-grained labels via distant supervision with the ESCO API. Our human evaluation shows that the distantly supervised labels give a signal of correctly annotated spans, where we achieve 41.3% accuracy on a large English label subset, and 70.4% accuracy on the Danish dev set, and 14.1% accuracy on the Danish test set. We manually correct the Danish test set with the correct labels from ESCO to create a gold

---

<sup>11</sup>We release the Danish anonymized raw data and annotations of the parts with permissible licenses from a governmental agency which is our collaborator. Links to our English data can be found at <https://github.com/kris927b/SkillSpan>. For anonymization, we perform it via manual annotation of job-related sensitive and personal data regarding Organization, Location, Contact, and Name following the work by Jensen et al. (2021a).

annotated set and keep the English labels as is, and thus silver labels.

Furthermore, domain adaptive pretraining helps to improve performance on the task specifically for English. The best performance is achieved with RemBERT on both the zero-shot setting ( $0.354_{\pm 0.021}$  weighted macro-F1) and few-shot setting ( $0.472_{\pm 0.014}$  weighted macro-F1), where they significantly outperform the other models. The strong performance is likely due to the pretraining data that contains both Danish and English.

Last, since the annotations are on the token level, this work can be extended to, for example, sequence labeling. We hope this dataset initiates further research in the area of skill classification.

## 6.7 Appendix

### 6.7.1 Data Statement KOMPETENCER

Following [Bender and Friedman \(2018\)](#), the following outlines the data statement for KOMPETENCER:

- A. **CURATION RATIONALE:** Collection of job postings in the English and Danish language for skill classification, to study the impact of skill changes from job postings.
- B. **LANGUAGE VARIETY:** The non-canonical data was collected from the StackOverflow job posting platform, an in-house job posting collection from our national labor agency collaboration partner (the Danish Agency for Labor Market and Recruitment<sup>12</sup>), and web extracted job postings from a large job posting platform. US (en-US), British (en-GB) English, and Danish (da-DK) are involved.
- C. **SPEAKER DEMOGRAPHIC:** Gender, age, race-ethnicity, socioeconomic status are unknown.
- D. **ANNOTATOR DEMOGRAPHIC:** Three hired project participants (age range: 25–30), gender: one female and two males, white European and Asian (non-Hispanic). Native language: Danish, Dutch. Socioeconomic status: higher-education students. Female annotator is a professional annotator with a background in Linguistics and the two males with a background in Computer Science.
- E. **SPEECH SITUATION:** Standard American, British English or Danish is used in job postings. Time frame of the data is between 2012–2021.
- F. **TEXT CHARACTERISTICS:** Sentences are from job postings posted on official job vacancy platforms.
- G. **RECORDING QUALITY:** N/A.
- H. **OTHER:** N/A.

---

<sup>12</sup><https://www.star.dk/en/>

PARAMETER	VALUE	RANGE
Optimizer	AdamW	
$\beta_1, \beta_2$	0.9, 0.99	
Dropout	0.2	0.1, 0.2, 0.3
Epochs	20	
Batch Size	32	
Learning Rate (LR)	1e-4	1e-3, 1e-4, 1e-5
LR scheduler	Slanted triangular	
Weight decay	0.01	
Decay factor	0.38	0.35, 0.38, 0.5
Cut fraction	0.2	0.1, 0.2, 0.3

Table 6.2: Hyperparameters of MACHAMP.

- I. PROVENANCE APPENDIX: The Danish job posting data is from our collaborators: The Danish Agency for Labour Market and Recruitment (STAR).

### 6.7.2 Reproducibility

We use the default hyperparameters in MACHAMP (van der Goot et al., 2021b) as shown in Table 6.2. For more details we refer to their paper. For the five random seeds we use 3477689, 4213916, 6828303, 8749520, and 9364029. All experiments with MACHAMP were ran on an NVIDIA<sup>®</sup> NVIDIA A100-SXM4 40GB GPU and an AMD<sup>®</sup> EPYC 7662 64-Core Processor.

### 6.7.3 Label Meaning

In Table 6.3, Table 6.4, and Table 6.5, we show the definitions of the labels.

### 6.7.4 Exact Results from Plots

In Table 6.6, we show the exact results of the plots from Figure 6.5 on English dev, English test, and Danish test respectively. In addition, we do

significance testing. Recently, the Almost Stochastic Order (ASO) test (Dror et al., 2019)<sup>13</sup> has been proposed to test statistical significance for deep neural networks over multiple runs. Generally, the ASO test determines whether a stochastic order (Reimers and Gurevych, 2018) exists between two models or algorithms based on their respective sets of evaluation scores. Given the single model scores over multiple random seeds of two algorithms  $\mathcal{A}$  and  $\mathcal{B}$ , the method computes a test-specific value ( $\epsilon_{\min}$ ) that indicates how far algorithm  $\mathcal{A}$  is from being significantly better than algorithm  $\mathcal{B}$ . When distance  $\epsilon_{\min} = 0.0$ , one can claim that  $\mathcal{A}$  stochastically dominant over  $\mathcal{B}$  with a predefined significance level. When  $\epsilon_{\min} < 0.5$  one can say  $\mathcal{A} \succeq \mathcal{B}$ . On the contrary, when we have  $\epsilon_{\min} = 1.0$ , this means  $\mathcal{B} \succeq \mathcal{A}$ . For  $\epsilon_{\min} = 0.5$ , no order can be determined. We compared all pairs of models based on five random seeds each using ASO with a confidence level of  $\alpha = 0.05$  (before adjusting for all pair-wise comparisons using the Bonferroni correction (Bonferroni, 1936)). Almost stochastic dominance ( $\epsilon_{\min} < 0.5$ ) is indicated in Figure 6.7 over all the splits.

### 6.7.5 Confusion Matrix Few-Shot

In Figure 6.8, we show the confusion matrix of the best performing few-shot model on the test set of the best run and investigate what the model does not predict correctly. Dissimilar from Figure 6.6, we only seem some confusion in the small cluster of S1-4. Giving the model a few Danish JPs substantially improved the prediction of A1, which relates to *attitudes* and gets predicted as S1: Communication, collaboration, and creativity.

---

<sup>13</sup>Implementation of Dror et al. (2019) can be found at <https://github.com/Kaleidophon/deep-significance> (Ulmer, 2021)

LABEL	SUBJECT	DEFINITION
0000	ARTIFACT	ARTIFACT
A1	Attitudes	Individual work styles that can affect how well someone performs a job.
A2	Values	Principles or standards of behavior, revealing one's judgment of what is important in life.
K00	Generic programmes and qualifications	Generic programmes and qualifications are those providing fundamental and personal skills education which cover a broad range of subjects and do not emphasise or specialise in a particular broad or narrow field.
K01	Education	NO-DEFINITION
K02	Arts and humanities	NO-DEFINITION
K03	Social sciences, journalism and information	NO-DEFINITION
K04	Business, administration and law	NO-DEFINITION
K05	Natural sciences, mathematics and statistics	NO-DEFINITION
K06	Information and communication technologies (icts)	NO-DEFINITION
K07	Engineering, manufacturing and construction not elsewhere classified	NO-DEFINITION
K08	Agriculture, forestry, fisheries and veterinary	NO-DEFINITION
K09	Health and welfare	NO-DEFINITION
K10	Services	NO-DEFINITION
K99	Field unknown	NO-DEFINITION
L1	Languages	Ability to communicate through reading, writing, speaking and listening in the mother tongue and/or in a foreign language.

**Table 6.3: Definition of ESCO Labels (Part 1).** Indicated are the definitions of the ESCO labels used in this work taken from the ESCO taxonomy. Artifacts of the ESCO API are K? and S?, and 0000, this means that no component was found.



LABEL	SUBJECT	DEFINITION
S1	Communication, collaboration and creativity	Communicating, collaborating, liaising, and negotiating with other people, developing solutions to problems, creating plans or specifications for the design of objects and systems, composing text or music, performing to entertain an audience, and imparting knowledge to others.
S2	Information skills	Collecting, storing, monitoring, and using information; Conducting studies, investigations and tests; maintaining records; managing, evaluating, processing, analysing and monitoring information and projecting outcomes.
S3	Assisting and caring	Providing assistance, nurturing, care, service and support to people, and ensuring compliance to rules, standards, guidelines or laws.
S4	Management skills	Managing people, activities, resources, and organisation; developing objectives and strategies, organising work activities, allocating and controlling resources and leading, motivating, recruiting and supervising people and teams.
S5	Working with computers	Using computers and other digital tools to develop, install and maintain ICT software and infrastructure and to browse, search, filter, organise, store, retrieve, and analyse data, to collaborate and communicate with others, to create and edit new content.

Table 6.4: **Definition of ESCO Labels (Part 2)**. Indicated are the definitions of the ESCO labels used in this work taken from the ESCO taxonomy. Artifacts of the ESCO API are K? and S?, and 0000, this means that no component was found.

LABEL	SUBJECT	DEFINITION
S6	Handling and moving	Sorting, arranging, moving, transforming, fabricating and cleaning goods and materials by hand or using handheld tools and equipment. Tending plants, crops and animals.
S7	Constructing	Building, repairing, installing and finishing interior and exterior structures.
S8	Working with machinery and specialised equipment	Controlling, operating and monitoring vehicles, stationary and mobile machinery and precision instrumentation and equipment.
K?	ARTIFACT	ARTIFACT
S?	ARTIFACT	ARTIFACT

Table 6.5: **Definition of ESCO Labels (Part 3)**. Indicated are the definitions of the ESCO labels used in this work taken from the ESCO taxonomy. Artifacts of the ESCO API are K? and S?, and 0000, this means that no component was found.

MODEL	EN DEV	EN TEST	DA TEST
BERT <sub>base</sub> (EN)	0.628 $\pm$ 0.004	0.632 $\pm$ 0.007	0.038 $\pm$ 0.008
JobBERT (EN)	0.628 $\pm$ 0.006	<b>0.644<math>\pm</math>0.006*</b>	0.063 $\pm$ 0.005
RemBERT (EN)	<b>0.629<math>\pm</math>0.003</b>	0.637 $\pm$ 0.007	0.354 $\pm$ 0.021
DaBERT (DA)	0.088 $\pm$ 0.013	0.076 $\pm$ 0.012	0.199 $\pm$ 0.058
DaJobBERT (DA)	0.101 $\pm$ 0.024	0.096 $\pm$ 0.024	0.395 $\pm$ 0.021
RemBERT (DA)	0.116 $\pm$ 0.052	0.098 $\pm$ 0.040	0.166 $\pm$ 0.141
RemBERT (EN+DA)	<b>0.629<math>\pm</math>0.006*</b>	0.643 $\pm$ 0.006	<b>0.472<math>\pm</math>0.014*</b>

Table 6.6: **Exact Results on Splits**. Indicated are the exact results of the bar plots in Figure 6.5. Significance tested with Almost Stochastic Order (Dror et al., 2019) test with Bonferroni correction (Bonferroni, 1936). Bold indicates highest average weighted macro-F1 and asterisk indicates significance.

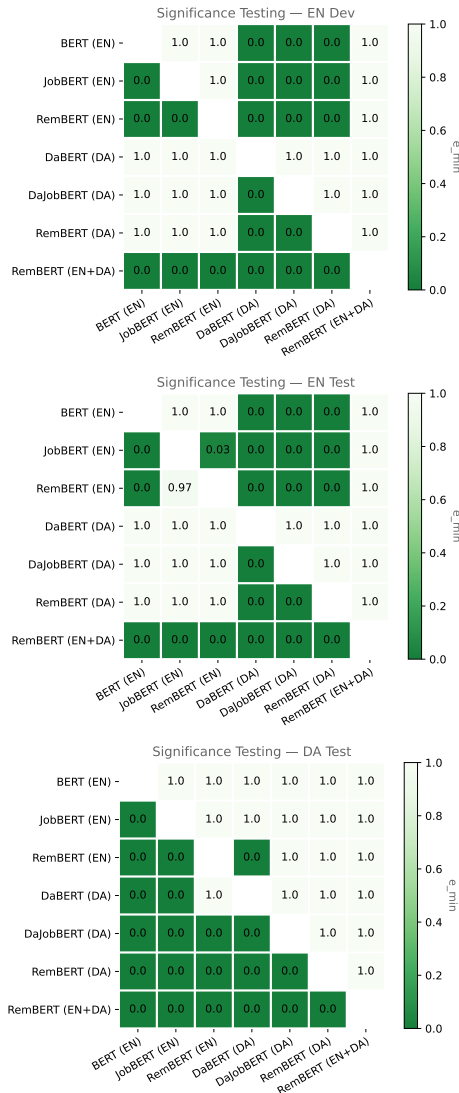


Figure 6.7: **Results Almost Stochastic Order.** ASO scores are expressed in  $\epsilon_{\min}$ . The significance level  $\alpha = 0.05$  is adjusted accordingly by using the Bonferroni correction (Bonferroni, 1936). Almost stochastic dominance ( $\epsilon_{\min} < 0.5$ ) is indicated in the colored boxes: On EN TEST, JobBERT is almost stochastically dominant over RemBERT (EN), with  $\epsilon_{\min} = 0.03$ .

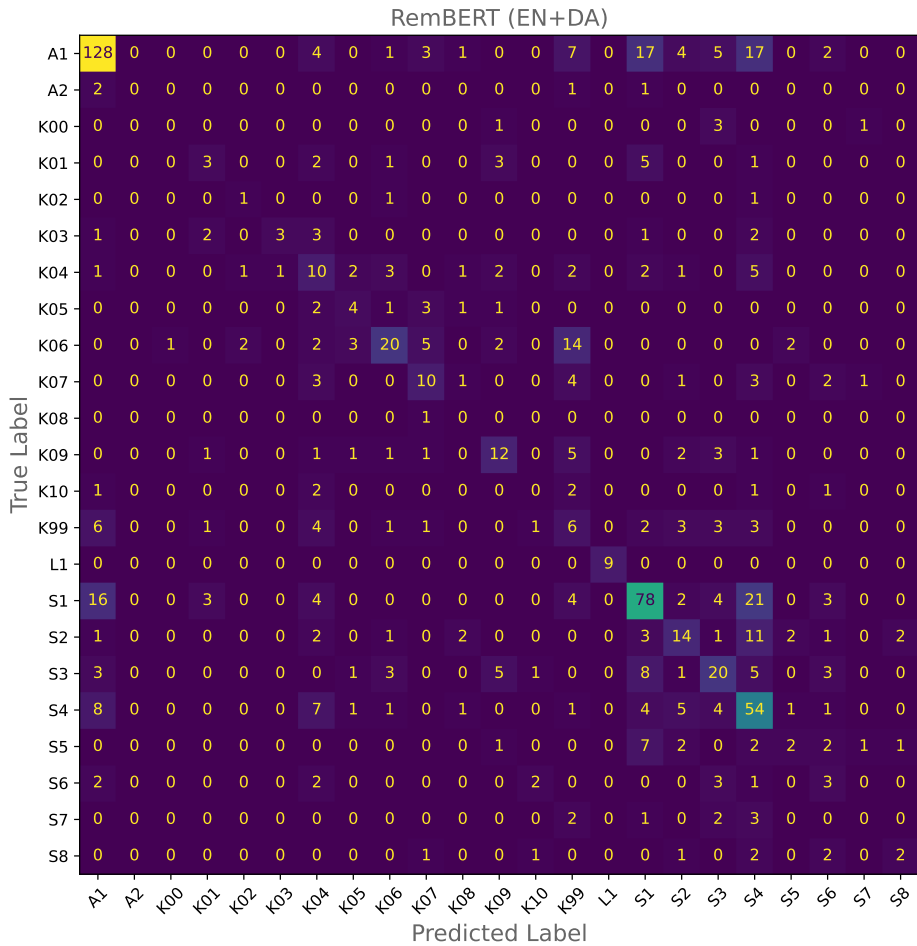


Figure 6.8: **Confusion Matrix of RemBERT (EN+DA)**. We show the confusion matrix of the few-shot setting with RemBERT (EN+DA). On the diagonal are the correctly predicted labels. There is less confusion in this model as compared to RemBERT (EN). We suspect the additional Danish data benefits the prediction of A1.

## **Part III**

# **Modeling Occupational Skills**



# Chapter 7

## Skill Extraction from Job Postings using Weak Supervision

The work presented in this chapter is based on a paper that has been published as: Mike Zhang, Kristian Nørgaard Jensen, Rob van der Goot, and Barbara Plank. Skill extraction from job postings using weak supervision. In *Proceedings of RecSysHR'22*. RecSysHR'22, 2022d. URL [https://ceur-ws.org/Vol-3218/RecSysHR2022-paper\\_10.pdf](https://ceur-ws.org/Vol-3218/RecSysHR2022-paper_10.pdf).

## Abstract

Aggregated data obtained from job postings provide powerful insights into labor market demands, and emerging skills, and aid job matching. However, most extraction approaches are supervised and thus need costly and time-consuming annotation. To overcome this, we propose Skill Extraction with Weak Supervision. We leverage the European Skills, Competences, Qualifications and Occupations taxonomy to find similar skills in job ads via latent representations. The method shows a strong positive signal, outperforming baselines based on token-level and syntactic patterns.

## 7.1 Introduction

The labor market is under constant development—often due to changes in technology, migration, and digitization—and so are the skill sets required (Brynjolfsson and McAfee, 2011, 2014). Consequentially, large quantities of job vacancy data is emerging on a variety of platforms. Insights from this data on labor market skill set demands could aid, for instance, job matching (Balog et al., 2012). The task of automatic *skill extraction* (SE) is to extract the competences necessary for any occupation from unstructured text.

Previous work on supervised SE frame it as a sequence labeling task (e.g., (Sayfullina et al., 2018; Tamburri et al., 2020; Chernova, 2020; Zhang et al., 2022c,a; Green et al., 2022; Gnehm et al., 2022b)) or multi-label classification (Bhola et al., 2020). Annotation is a costly and time-consuming process with little annotation guidelines to work with. This could be alleviated by using predefined skill inventories.

In this work, we approach span-level SE with weak supervision: We leverage the European Skills, Competences, Qualifications and Occupations (ESCO; (le Vrang et al., 2014)) taxonomy and find similar spans that relate to ESCO skills in embedding space (Figure 7.1). The advantages are twofold: First, labeling skills becomes obsolete, which mitigates the cumbersome process of annotation. Second, by extracting skill phrases, this



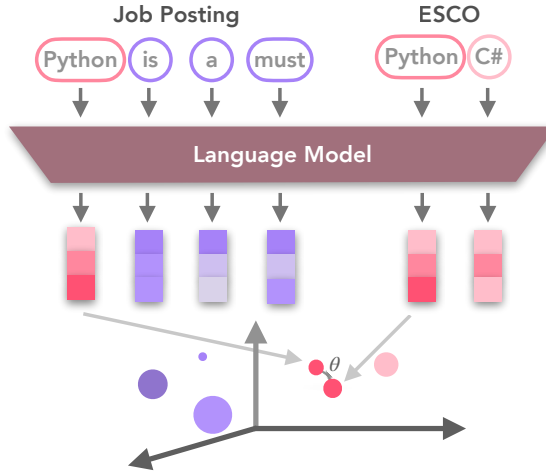


Figure 7.1: **Weakly Supervised Skill Extraction.** All ESCO skills and n-grams are extracted and embedded through a language model, e.g., RoBERTa (Liu et al., 2019), to get representations. We label *spans* from job postings close in vector space to the ESCO skill.

could possibly enrich skill inventories (e.g., ESCO) by finding paraphrases of existing skills. We seek to answer: *How viable is Weak Supervision in the context of SE?* We contribute: ① A novel weakly supervised method for SE; ② A linguistic analysis of ESCO skills and their presence in job postings; ③ An empirical analysis of different embedding pooling methods for SE for two skill-based datasets.<sup>1</sup>

## 7.2 Methodology

Formally, we consider a set of job postings  $\mathcal{D}$ , where  $d \in \mathcal{D}$  is a set of sequences (e.g., job posting sentences) with the  $i^{\text{th}}$  input sequence  $\mathcal{T}_d^i = [t_1, t_2, \dots, t_n]$  and a target sequence of BIO-labels  $\mathcal{Y}_d^i = [y_1, y_2, \dots, y_n]$  (e.g., “B-SKILL”, “I-SKILL”, “O”).<sup>2</sup> The goal is to use an algorithm, which predicts skill spans by assigning an output label sequence  $\mathcal{Y}_d^i$  for each token sequence  $\mathcal{T}_d^i$  from a job posting based on representational similarity of a

<sup>1</sup><https://github.com/jjzha/skill-extraction-weak-supervision>

<sup>2</sup>Definition of labels can be found in (Zhang et al., 2022a).

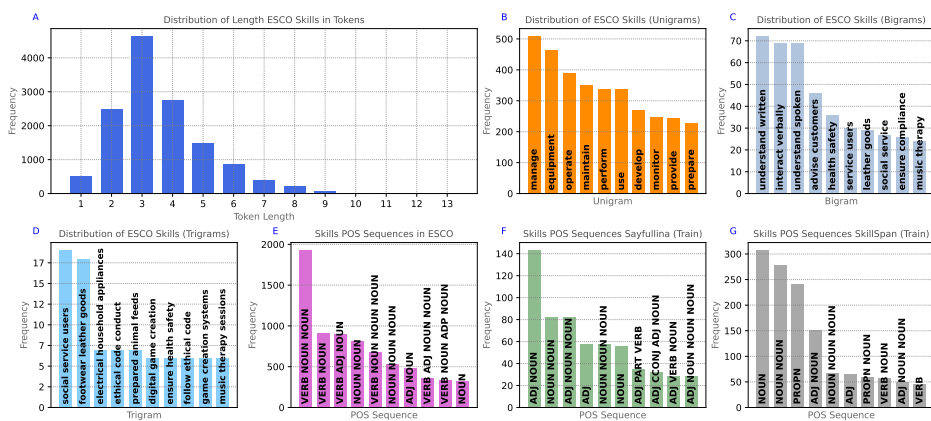


Figure 7.2: **Surface-level Statistics of ESCO.** We show various statistics of ESCO. (A) ESCO skills token length, the mode is three tokens. (B) Most frequent unigrams of ESCO skills. (C) Most frequent bigrams of ESCO skills. (D) Most frequent trigrams of ESCO skills. (E) Most frequent POS sequences of ESCO skills. Last, we show the POS sequences of unique skills in both train sets of *Sayfullina* and *SkillSpan* (F–G).

span to any skill in ESCO.

### 7.2.1 Data

We use the datasets from (Zhang et al., 2022a) (*SkillSpan*) and a modification of (Sayfullina et al., 2018) (*Sayfullina*).<sup>3</sup> In Table 7.1, we show the statistics of both. *SkillSpan* contain nested labels for skill and knowledge components (le Vrang et al., 2014). To make it fit for our weak supervision approach, we simplify their dataset by considering both skills and knowledge labels as one label (i.e., B-KNOWLEDGE becomes B-SKILL).

<sup>3</sup>In contrast to *SkillSpan*, *Sayfullina* has a skill in every sentence, where they focus on categorizing sentences for soft skills.

	Statistics	<i>Sayfullina</i>	<i>SkillSpan</i>
TRAIN	# Sentences	3,703	5,866
	# Tokens	53,095	122,608
	# Skill Spans	3,703	3,325
DEV.	# Sentences	1,856	3,992
	# Tokens	26,519	52,084
	# Skill Spans	1,856	2,697
TEST	# Sentences	1,848	4,680
	# Tokens	26,569	57,528
	# Skill Spans	1,848	3,093
	<b>Avg. Len. Skills</b>	1.77	2.92

Table 7.1: **Statistics of Datasets.** Indicated is each dataset and their respective number of sentences, tokens, skill spans, and the average length of skills in tokens.

### 7.2.2 ESCO Statistics

We use ESCO as a weak supervision signal for discovering skills in job postings. There are 13,890 ESCO skills.<sup>4</sup> In Figure 7.2, we show statistics of the taxonomy: (A) On average most skills are 3 tokens long. In (C-D), we show n-grams frequencies with range [1; 3]. We can see that the most frequent uni- and bigrams are verbs, while the most frequent trigrams consist of nouns.

Additionally, we show an analysis of ESCO skills from a linguistic perspective. We tag the training data using the publicly available MaChAmp v0.2 model (van der Goot et al., 2021b) trained on all Universal Dependencies 2.7 treebanks (Nivre et al., 2017).<sup>5</sup> Then, we count the most frequent Part-of-Speech (POS) tags in all sources of data (E-G). ESCO’s most frequent tag sequences are VERB-NOUN, these are not as frequent in *Sayfullina* nor *SkillSpan*. *Sayfullina* mostly consists of adjectives, which is

<sup>4</sup>Per 25-03-2022, taking ESCO v1.0.9.

<sup>5</sup>A Udify-based (Kondratyuk and Straka, 2019) multi-task model for POS, lemmatization, dependency parsing, built on top of the transformers library (Wolf et al., 2020a), and specifically using mBERT (Devlin et al., 2019b).

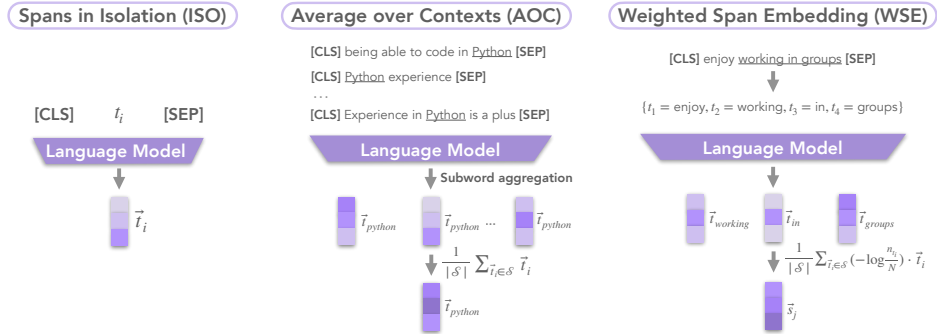


Figure 7.3: **Skill Representations.** We show different methods to embed ESCO skill phrases. The approaches are inspired by [Litschko et al. \(2022\)](#). We embed a skill by encoding it directly without surrounding context (**left**). We aggregate different contextual representations of the same skill term (**middle**). Last, we encode the skill phrase via a weighted sum of embeddings with each token’s inverse document frequency as weight (**right**). For the middle and right methods,  $\mathcal{S}$  is the number of sentences where the ESCO skill appears.

attributed to the categorization of soft skills. *SkillSpan* mostly consists of NOUN sequences. Overall, we observe most skills consist of verb and noun phrases.

### 7.2.3 Baselines

As our approach is to find similar n-grams based on ESCO skills, we choose an n-gram range of  $[1; 4]$  (where 4 is the median) derived from [Figure 7.2 \(A\)](#). For higher matching probability, we apply an additional pre-processing step to the ESCO skills by removing non-tokens (e.g., brackets) and words between brackets (e.g., “Java (programming)” becomes “Java”). We have three baselines:

**Exact Match:** We do exact substring matching with ESCO and the sentences in both datasets.

**Lemmatized Match:** ESCO skills are written in the infinitive form. We take the same approach as exact match on the training sets, now with

---

**Algorithm 3: Weakly Supervised Skill Extraction**

---

```

1  $M \leftarrow \{\text{RoBERTa}, \text{JobBERT}\};$ 
2  $E \leftarrow \{\text{ISO}, \text{AOC}, \text{WSE}\};$ 
3  $\tau \leftarrow [0, 1];$ 
4  $D$  // A set of sentences from job postings;
5  $L \leftarrow \emptyset;$ 
6  $P \leftarrow D$  // A set of sentences from job postings;
7  $S \leftarrow S_E$  // ESCO Skill embeddings of type  $E$ ;
8 for  $p \in P$  do
9    $\theta \leftarrow 0;$ 
10  for  $n \in \text{split}(p)$  do
11    // Each ngram  $n$  of size 1 – 4  $E \leftarrow M(n);$ 
12     $\Theta \leftarrow \text{CosSim}(S, E);$ 
13    if  $\max(\Theta) > \tau \wedge \max(\Theta) > \theta$  then
14       $\theta \leftarrow \max(\Theta);$ 
15    end
16  end
17   $L \leftarrow L \cup \{\theta\};$ 
18 end
19 return  $L;$ 

```

---

the lemmatized data of both. The data is lemmatized with MaChAmp v0.2 (van der Goot et al., 2021b).

**POS Sequence Match:** Motivated by the observation that certain POS sequences often overlap between sources (Figure 7.2, E-G), we attempt to match POS sequences within ESCO with the POS sequences in the datasets. For example NOUN-NOUN, NOUN, VERB-NOUN and ADJ-NOUN sequences are commonly occurring in all three sources.

### 7.2.4 Skill Representations

We investigate several encoding strategies to match n-gram representations to embedded ESCO skills, the approaches are inspired by Litschko et al. (2022), where they applied them to Information Retrieval. The language models (LMs) used to encode the data are RoBERTa (Liu et al., 2019)

and the domain-specific JobBERT (Zhang et al., 2022a). All obtained vector representations of skill phrases with the three previous encoding methods are compared pairwise with each n-gram created from *Sayfullina* and *SkillSpan*. An explanation of the methods (see Figure 7.3):

**Span in Isolation (ISO):** We encode skill phrases  $t$  from ESCO in isolation using the aforementioned LMs, without surrounding contexts.

**Average over Contexts (AOC):** We leverage the surrounding context of a skill phrase  $t$  by collecting all the sentences containing  $t$ . We use all available sentences in the job postings dataset (excluding TEST). For a given job posting sentence, we encode  $t$  by using one of the previous mentioned LMs. We average the embeddings of its constituent subwords to obtain the final embedding  $t$ .

**Weighted Span Embedding (WSE):** We obtain all inverse document frequency (idf) values of each token  $t_i$  via

$$\text{idf} = -\log \frac{n_{t_i}}{N},$$

where  $n_{t_i}$  is the number of occurrences of  $t_i$  and  $N$  the total number of tokens in our dataset. We encode the input sentence and compute the weighted sum of the embeddings ( $\vec{s}_j$ ) of the specific skill phrase in the sentence, where each  $t_i$ 's IDF scores are used as weights. Again, we only use the first subword token for each tokenized word. Formally, this is

$$\vec{s}_j = \sum_{\vec{t}_i} \left(-\log \frac{n_{t_i}}{N}\right) \cdot \vec{t}_i.$$

### 7.2.5 Matching

We rank pairs of ESCO embeddings  $\vec{t}$  and encoded candidate n-grams  $\vec{g}$  in decreasing order of cosine similarity (CosSim), calculated as

$$\text{CosSim}(\vec{t}, \vec{g}) = \frac{\vec{t}^T \vec{g}}{\|\vec{t}\| \|\vec{g}\|}.$$

We show our pseudocode of the matching algorithm in Algorithm 3.

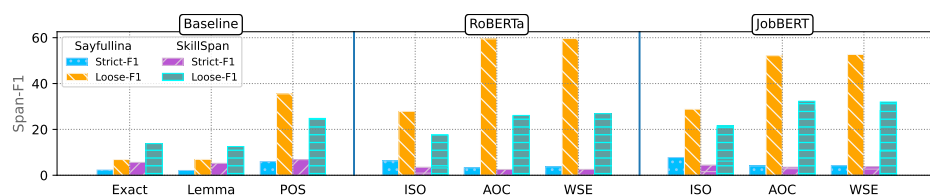


Figure 7.4: **Results of Methods.** Results on *Sayfullina* and *SkillSpan* are indicated by “Baseline” showing performance of Exact, Lemmatized (Lemma), and Part-of-Speech (POS). The performance of ISO, AOC, and WSE are separated by model, indicated by “RoBERTa” and “JobBERT”. The performance of RoBERTa and JobBERT on *SkillSpan* is determined by the best performing CosSim threshold (0.8).

Note that in *SkillSpan* we have to set a threshold for CosSim, as there are sentences with no skills. A threshold allows us to have a “no skill” option. As seen in Figure 7.5, Subsection 7.5.1 the threshold sensitivity on *SkillSpan* differs for JobBERT: Performance fluctuates, compared to RoBERTa. Precision goes up with a higher threshold, while recall goes down. For RoBERTa, it stays similar until CosSim= 0.9. We use CosSim= 0.8 as over 2 LMs and 3 methods it provides the best cutoff.

## 7.3 Analysis of Results

### 7.3.1 Results

Our main results (Figure 7.4) show the baselines against ISO, AOC, and WSE of both datasets. We evaluate with two types of F1, following van der Goot et al. (2021a): strict and loose-F1. For full model fine-tuning, RoBERTa achieves 91.31 and 98.55 strict and loose F1 on *Sayfullina* respectively. For *SkillSpan*, this is 23.21 and 44.72 strict and loose F1 (on the available subsets of *SkillSpan*). JobBERT achieves 90.18 and 98.19 strict and loose F1 on *Sayfullina*, 49.44 and 74.41 strict and loose F1 on *SkillSpan*. The large difference between results is most likely due to lack of negatives in *Sayfullina*, i.e., all sentences contain a skill, which makes the

	Gold	Predicted
<i>Sayfullina</i>	...a dynamic customer focused person to join... ...strong leadership and team management skills ... ...speak and written english skills ... ...a team environment and working independently skills ... ...tangible business benefit extremely articulate and...	...a dynamic customer focused person to join... ...strong leadership and team management skills ... ...speak and written english skills... ...a team environment and working independently skills... ...tangible business benefit extremely articulate and...
<i>SkillSpan</i>	...researcher within machine learning and sensory system design ... ...standards and procedures accessing and updating records ... ...with a passion for education to... ...understands Agile as a mindset... ...experience with AWS GCP Microsoft Azure...	...researcher within machine learning and sensory system design... ...standards and procedures accessing and updating records... ...with a passion for education to... ...understands Agile as a mindset... ...experience with AWS GCP Microsoft...

**Table 7.2: Qualitative Examples of Predicted Spans.** We show the gold versus predicted spans of the best performing model on both datasets. The first 5 qualitative examples are from *Sayfullina* (RoBERTa with WSE), the last 5 are from *SkillSpan*. Yellow the gold span and pink indicates the predicted span. The examples show many partial overlaps with the gold spans (but also incorrect ones), hence the high loose-F1.

task easier. These results highlight the difficulty of SE on *SkillSpan*, where there are negatives as well (sentences with no skills).

The exact match baseline on *SkillSpan* is higher than *Sayfullina*. We attribute this to *SkillSpan* also containing “hard skills” (e.g., “Python”), which is easier to match substrings with than “soft skills”.<sup>6</sup>

For the performance of the skill representations on *Sayfullina*, RoBERTa and JobBERT outperform the Exact and Lemmatized baseline on strict-F1. For the POS baseline, only the ISO method of both models is slightly better. JobBERT performs better than RoBERTa in strict-F1 on both datasets.

There is a substantial difference between strict and loose-F1 on both datasets. This indicates that there is partial overlap among the predicted and gold spans. RoBERTa performs best for *Sayfullina*, achieving 59.61 loose-F1 with WSE. In addition, the best performing method for JobBERT is also WSE (52.69 loose-F1). For *SkillSpan* we see a drop, JobBERT outperforms RoBERTa with AOC (32.30 vs. 26.10 loose-F1) given a threshold of  $\text{CosSim} = 0.8$ . We hypothesize this drop in performance compared to *Sayfullina* could be attributed again to *SkillSpan* containing negative examples as well (i.e., sentences with no skill).

<sup>6</sup>The exact numbers (+precision and recall) are in Table 7.3, Subsection 7.5.1, including the definition of strict and loose-F1.



### 7.3.2 Qualitative Analysis

A qualitative analysis (Table 7.2) reveals there is strong partial overlap with gold vs. predicted spans on both datasets, e.g., “...strong leadership and team management skills...” vs. “...strong leadership and team management skills...”, indicating the viability of this method.

## 7.4 Conclusion

We investigate whether the ESCO skill taxonomy suits as weak supervision signal for Skill Extraction. We apply several skill representation methods based on previous work. We show that using representations of ESCO skills can aid us in this task. We achieve high loose-F1, indicating there is partial overlap between the predicted and gold spans, but need refined off-set methods to get the correct span out (e.g., human post-editing or automatic methods such as candidate filtering). Nevertheless, we see this approach as a strong alternative for supervised Skill Extraction from job postings.

Future work could include going towards multilingual Skill Extraction, as ESCO consists of 27 languages, exact matching should be trivial. For the other methods several considerations need to be taken into account, e.g., a POS-tagger and/or lemmatizer for another language and a language-specific model.

	Dataset → ↓ Method, Metric →	SAYFULLINA						SKILLSPAN					
		Strict (P   R   F1)			Loose (P   R   F1)			Strict (P   R   F1)			Loose (P   R   F1)		
Baseline	Exact	9.27	1.30	2.28	25.48	3.95	6.84	23.82	3.21	5.62	43.68	8.27	13.79
	Lemmatized	8.49	1.19	2.09	25.87	4.00	6.93	23.90	2.97	5.21	41.09	7.49	12.52
	POS	5.99	5.95	5.97	36.55	34.51	35.50	5.97	7.88	6.79	19.34	34.71	24.80
RoBERTa	ISO	6.26	6.25	6.26	26.90	28.98	27.90	2.90	4.24	3.43	12.69	28.61	17.56
	AOC	3.24	3.24	3.24	64.04	55.53	59.48	2.23	2.93	2.53	20.08	37.56	26.10
	WSE	3.67	3.67	3.67	64.64	55.32	59.61	2.29	2.93	2.57	20.90	37.79	26.85
JobBERT	ISO	7.71	7.72	7.71	27.76	29.95	28.82	4.17	4.65	4.39	17.07	29.48	21.61
	AOC	4.04	4.05	4.05	56.50	48.41	52.14	4.44	2.96	3.54	33.64	31.28	32.30
	WSE	4.15	4.16	4.15	56.98	49.00	52.69	4.78	3.08	3.74	34.01	30.33	31.95

Table 7.3: We show the exact numbers of the performance of the methods.

## 7.5 Appendix

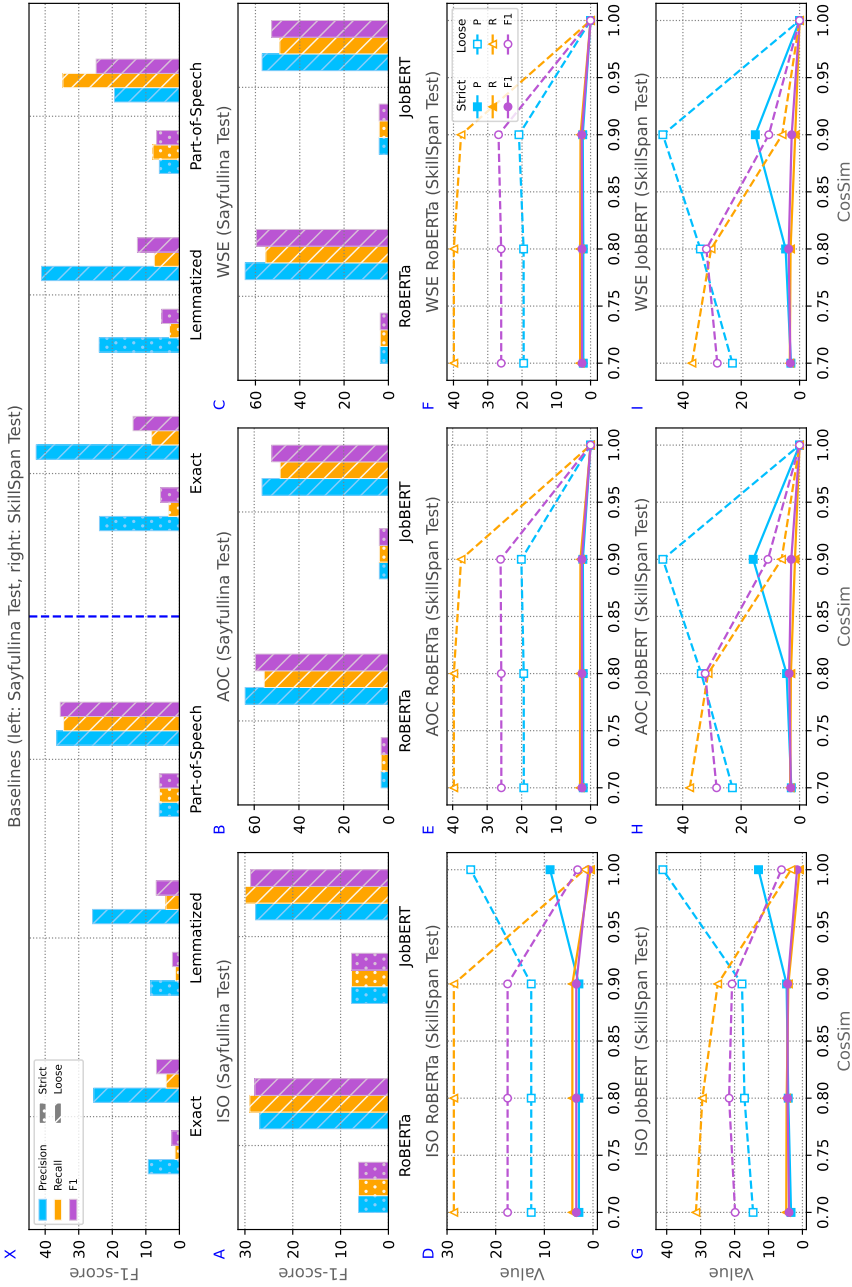


Figure 7.5: **Results of Methods.** Results of the baselines are in (X), the performance of ISO, AOC, and WSE on *Sayfullina* in (A-C), and the same performance on *SkillSpan* in (D-I) based on the model (RobERTa or JobBERT). In D-F, we show the precision (P), recall (R), and F1 differences when taking an increasing CosSim.

### 7.5.1 Exact Results

### 7.5.2 Definition F1

As mentioned, we evaluate with two types of F1-scores, following [van der Goot et al. \(2021a\)](#). The first type is the commonly used span-F1, where only the correct span and label are counted towards true positives. This is called `strict-F1`. In the second variant, we seek for partial matches, i.e., overlap between the predicted and gold span including the correct label, which counts towards true positives for precision and recall. This is called `loose-F1`. We consider the loose variant as well, because we want to analyze whether the span is “almost correct”.

### 7.5.3 Exact Numbers Results

We show the exact numbers of [Figure 7.4](#) in [Table 7.3](#) and more detailed results in [Figure 7.5](#). Results show that there is high precision among the baseline approaches compared to recall. This is balanced using the representation methods for *Sayfullina*. However, we observe that there is much higher recall for *SkillSpan* than precision.

# Chapter 8

## ESCOXLM-R: Multilingual Taxonomy-driven Pre-training for the Job Market Domain

The work presented in this chapter is based on a paper that has been published as: Mike Zhang, Rob van der Goot, and Barbara Plank. ESCOXLM-R: Multilingual Taxonomy-driven Pre-training for the Job Market Domain. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11871–11890, Toronto, Canada, 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.662. URL <https://aclanthology.org/2023.acl-long.662>.

## Abstract

The increasing number of benchmarks for Natural Language Processing (NLP) tasks in the computational job market domain highlights the demand for methods that can handle job-related tasks such as skill extraction, skill classification, job title classification, and de-identification. While some approaches have been developed that are specific to the job market domain, there is a lack of generalized, multilingual models and benchmarks for these tasks. In this study, we introduce a language model called ESCOXML-R, based on XLM-R<sub>large</sub>, which uses domain-adaptive pre-training on the European Skills, Competences, Qualifications and Occupations (ESCO) taxonomy, covering 27 languages. The pre-training objectives for ESCOXML-R include dynamic masked language modeling and a novel additional objective for inducing multilingual taxonomical ESCO relations. We comprehensively evaluate the performance of ESCOXML-R on 6 sequence labeling and 3 classification tasks in 4 languages and find that it achieves state-of-the-art results on 6 out of 9 datasets. Our analysis reveals that ESCOXML-R performs better on short spans and outperforms XLM-R<sub>large</sub> on entity-level and surface-level span-F1, likely due to ESCO containing short skill and occupation titles, and encoding information on the entity-level.

## 8.1 Introduction

The dynamic nature of labor markets, driven by technological changes, migration, and digitization, has resulted in a significant amount of job advertisement data (JAD) being made available on various platforms to attract qualified candidates (Brynjolfsson and McAfee, 2011, 2014; Balog et al., 2012). This has led to an increase in tasks related to JAD, including skill extraction (Kivimäki et al., 2013; Zhao et al., 2015; Sayfullina et al., 2018; Smith et al., 2019; Tamburri et al., 2020; Shi et al., 2020; Chernova, 2020; Bhola et al., 2020; Zhang et al., 2022a,c,d; Green et al., 2022; Gnehm et al., 2022b; Beauchemin et al., 2022; Decorte et al., 2022; Goyal et al., 2023), skill classification (Decorte et al., 2022; Zhang et al., 2022c),

job title classification (Javed et al., 2015, 2016; Decorte et al., 2021; Green et al., 2022), de-identification of entities in job postings (Jensen et al., 2021b), and multilingual skill entity linking (ESCO, 2022).

While some previous studies have focused on JAD in non-English languages (Zhang et al., 2022c; Gnehm et al., 2022b; Beauchemin et al., 2022), their baselines have typically relied on language-specific models, either using domain-adaptive pre-training (DAPT; Gururangan et al., 2020b) or off-the-shelf models. The lack of comprehensive, open-source JAD data in various languages makes it difficult to fully pre-train a language model (LM) using such data. In this work, we seek external resources that can help improve the multilingual performance on the JAD domain. We use the ESCO taxonomy (le Vrang et al., 2014), which is a standardized system for describing and categorizing the skills, competences, qualifications, and occupations of workers in the European Union. The ESCO taxonomy, which has been curated by humans, covers over 13,000 skills and 3,000 occupations in 27 languages. Therefore, we seek to answer: *To what extent can we leverage the ESCO taxonomy to pre-train a domain-specific and language-agnostic model for the computational job market domain?*

In this work, we release the first multilingual JAD-related model named ESCOXML-R, a language model based on XLM-R<sub>large</sub> that incorporates data from the ESCO taxonomy through the use of two pre-training objectives (Figure 8.1): Masked Language Modeling (MLM) and a novel ESCO relation prediction task (Section 8.2). We evaluate ESCOXML-R on 9 JAD-related datasets in 4 different languages covering 2 NLP tasks (Section 8.3). Our results show that ESCOXML-R outperforms previous state-of-the-art (SOTA) on 6 out of 9 datasets (Section 8.4). In addition, our fine-grained analysis reveals that ESCOXML-R performs better on short spans compared to XLM-R<sub>large</sub>, and consistently outperforms XLM-R<sub>large</sub> on entity-level and surface-level span-F1 (Section 8.5).

**Contributions.** In this work, we present and release the following:

- ESCOXML-R, an XLM-R<sub>large</sub>-based model, which utilizes domain-

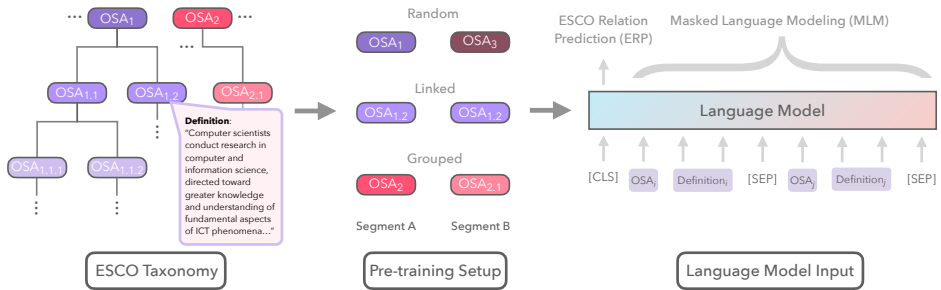


Figure 8.1: **ESCO Pre-training Objective:** From left to right, the figure illustrates the hierarchical structure of the ESCO taxonomy, which consists of occupations, skills, and aliases (OSA). Each OSA includes a definition. For the purposes of this study, we consider aliases of occupations to have the same definition as the occupation itself. In the middle of the figure, we show our pre-training setup. Pre-training instances are uniformly sampled in three ways: randomly, linked, or grouped (this is defined in Subsection 8.2.4). The selected instances (can be in different languages) are then fed to the language model, along with its description. We have two pre-training objectives: the regular MLM objective, and a new ESCO relation prediction objective, in which the goal is to predict which group the sampled instances belong to (Random, Linked, or Grouped).

adaptive pre-training on the 27 languages from ESCO.<sup>1</sup>

- The largest JAD evaluation study to date on 3 job-related tasks, comprising 9 datasets in 4 languages and 4 models.
- A fine-grained analysis of ESCOXML-R’s performance on different span lengths, and emerging entities (i.e., recognition of entities in the long tail).

<sup>1</sup>The code for ESCOXML-R is available as open-source: <https://github.com/mainlp/escoxlmr>. We further release ESCOXML-R under an Apache License 2.0 on HuggingFace: <https://huggingface.co/jjzha/esco-xml-roberta-large>.



## 8.2 ESCOXML-R

### 8.2.1 Preliminaries

In the context of pre-training, an LM is trained using a large number of unlabeled documents,  $\mathcal{X} = X^{(i)}$ , and consists of two main functions:  $f_{\text{encoder}}(\cdot)$ , which maps a sequence of tokens  $X = (x_1, x_2, \dots, x_t)$  to a contextualized vector representation for each token, represented as  $(h_1, h_2, \dots, h_t)$ , and  $f_{\text{head}}(\cdot)$ , the output layer that takes these representations and performs a specific task, such as pre-training in a self-supervised manner or fine-tuning on a downstream application. For example, BERT (Devlin et al., 2019b) is pre-trained using two objectives: MLM and Next Sentence Prediction (NSP). In MLM, a portion of tokens in a sequence  $X$  is masked and the model must predict the original tokens from the masked input. In the NSP objective, the model takes in two segments  $(X_A, X_B)$  and predicts whether segment  $X_B$  follows  $X_A$ . RoBERTa (Liu et al., 2019) is a variation of BERT that uses dynamic MLM, in which the masking pattern is generated each time a sequence is fed to the LM, and does not use the NSP task.

### 8.2.2 Multilinguality

Both BERT and RoBERTa have been extended to support multiple languages, resulting in multilingual BERT (mBERT; Devlin et al., 2019b) and XLM-RoBERTa (XLM-R; Conneau et al., 2020). XLM-R was found to outperform mBERT on many tasks (e.g., Conneau et al., 2020; Hu et al., 2020; Lauscher et al., 2020) due to careful tuning, sampling, and scaling to larger amounts of textual data. Because of this, our ESCOXML-R model is based on XLM-R<sub>large</sub>.

### 8.2.3 European Skills, Competences, Qualifications and Occupations Taxonomy

The European Skills, Competences, Qualifications, and Occupations (ESCO; le Vrang et al., 2014) taxonomy is a standardized system for

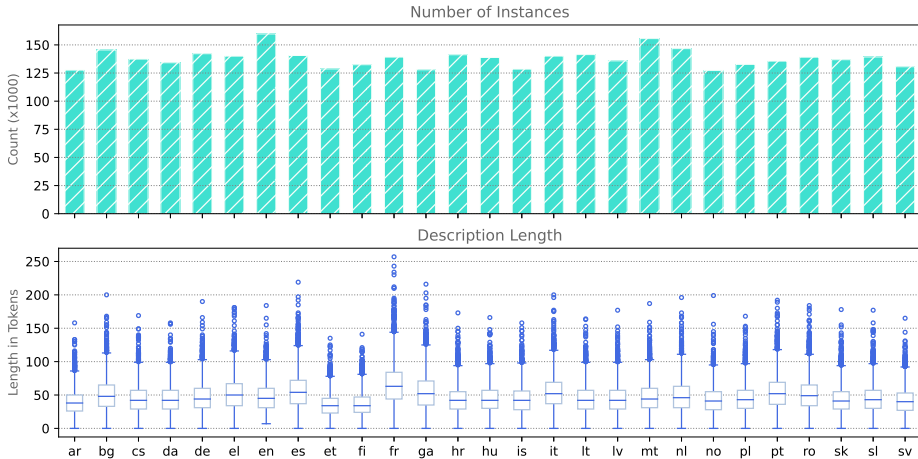


Figure 8.2: **Statistics of Pre-training Data.** The ESCO dataset contains descriptions in 27 languages, with a combined total of approximately 3.72 million descriptions (i.e., instances). On average, there are around 130,000 descriptions per language. The average length of each description is 26.3 tokens, with some descriptions reaching a maximum length of 150 or more tokens, as shown by the outliers in the boxplot.

describing and categorizing the skills, competences, qualifications, and occupations of workers in the European Union (EU). It is designed to serve as a common language for the description of skills and qualifications across the EU, facilitating the mobility of workers by providing a common reference point for the recognition of qualifications and occupations. The taxonomy is developed and maintained by the European Commission and is based on the International Classification of Occupations and the International Standard Classification of Education. It includes 27 European languages: Bulgarian (ar), Czech (cs), Danish (da), German (de), Greek (el), English (en), Spanish (es), Estonian (et), Finnish (fi), French (fr), Gaelic (ga), Croatian (hr), Hungarian (hu), Icelandic (is), Italian (it), Lithuanian (lt), Latvian (lv), Maltese (mt), Dutch (nl), Norwegian (no), Polish (pl), Portuguese (pt), Romanian (ro), Slovak (sk), Slovenian (sl), Swedish (sv), and Arabic (ar). Currently, it describes 3,008 occupations

and 13,890 skills/competences (SKC) in all 27 languages.<sup>2</sup>

The ESCO taxonomy includes a hierarchical structure with links between occupations, skills, and aliases (OSA). In this work, we focus on the occupation pages and extract the following information from the taxonomy:<sup>3</sup>

- **ESCO Code:** The taxonomy code for the specific occupation or SKC.
- **Occupation Label:** The preferred occupation name (i.e., title of the occupation).
- **Occupation Description/Definition:** A description of the responsibilities of the specific occupation.
- **Major Group Name:** The name of the overarching group to which the occupation belongs, e.g., “Veterinarians” for the occupation “animal therapist”.
- **Alternative Labels:** Aliases for the specific occupation, e.g., “animal rehab therapist” for the occupation “animal therapist”.
- **Essential Skills:** All necessary SKCs for the occupation, including descriptions of these.
- **Optional Skills:** All optional SKCs for the occupation, including descriptions of these.

In [Figure 8.2](#), we present the distribution of pre-training instances and the mean description lengths for each language in the ESCO taxonomy. Note that the number of descriptions is not the same for all languages, and we do not count empty descriptions (i.e., missing translations) for certain occupations or SKCs.

---

<sup>2</sup>Note that ESCO now also includes Ukrainian, but this model was trained before that inclusion. We use the ESCO V1.0.9 API to extract the data. ESCO contains an Apache 2.0 and a European Union Public License 1.2.

<sup>3</sup>An example of the extracted information can be found in [Listing 1 \(Subsection 8.8.1\)](#), and the original page can be accessed at <https://bit.ly/3DY1zsX>.

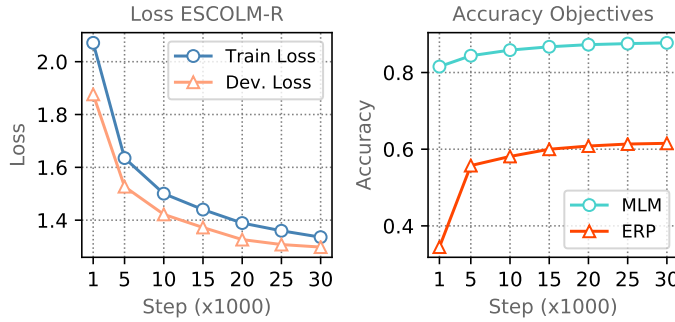


Figure 8.3: **Pre-Training Statistics.** The final log loss for the training set is 1.34, while the log loss for the development set is 1.30. The MLM accuracy is 84.3%, while the Entity Relationship Prediction (ERP) accuracy is 60.0%. These results were obtained after approximately 1.04 epochs of training on the total data.

### 8.2.4 Pre-training Setup

To improve our  $\text{XLM-R}_{\text{large}}$ -based model, we employ domain-adaptive pre-training techniques as described in previous work such as [Alsentzer et al. \(2019b\)](#); [Han and Eisenstein \(2019\)](#); [Lee et al. \(2020b\)](#); [Gururangan et al. \(2020b\)](#); [Nguyen et al. \(2020b\)](#). Given the limited amount of training data (3.72M sentences), we utilize the  $\text{XLM-R}_{\text{large}}$  checkpoint provided by the HuggingFace library ([Wolf et al., 2020a](#)) as a starting point.<sup>4</sup> Our aim is to fine-tune the model to internalize domain-specific knowledge related to occupation and SKCs, while maintaining its general knowledge acquired during the original pre-training phase.

We introduce a novel self-supervised pre-training objective for ESCOXML-R, inspired by LinkBERT from [Yasunaga et al. \(2022a\)](#). We view the ESCO taxonomy as a graph of occupations and SKCs ([Figure 8.1](#)), with links between occupations or occupations and SKCs in various languages. By placing similar occupations or SKCs in the same context window and in different languages, we can learn from the links between (occupation  $\leftrightarrow$  occupation) and (occupation  $\leftrightarrow$  SKCs) in different languages for true

<sup>4</sup><https://huggingface.co/xlm-roberta-large>

cross-lingual pre-training. In addition to the MLM pre-training objective, which is used to learn concepts within contexts, we introduce another objective called ESCO Relation Prediction (ERP) to internalize knowledge of connections within the taxonomy in the LM. We take an anchor concept ( $C_A$ ) by concatenating it with its description ( $X_A$ ) from the ESCO taxonomy and sample an additional concept ( $C_B$ ) concatenated with its description ( $X_B$ ) to create LM input [CLS]  $C_A X_A$  [SEP]  $C_B X_B$  [SEP].<sup>5</sup> We sample  $C_B X_B$  in three ways with uniform probability:

1. *Random*: We randomly sample  $C_B X_B$  from the ESCO taxonomy, in any language;
2. *Linked*: We sample  $C_B X_B$  in any language from the same occupation page, for example, an “animal therapist” (or an alias of the “animal therapist”, e.g., “animal rehab therapist”) should have knowledge of “animal behavior”;
3. *Grouped*: We sample  $C_B X_B$  from the same major group in any language. For the same example “animal therapist”, it comes from major group 2: Professionals → group 22: Health professionals. Several other concepts, e.g., “Nursing professionals” fall under this major group.

### 8.2.5 Pre-training Objectives

The LM is trained using two objectives. First is the MLM objective, and the second is the ERP objective, where the task is to classify the relation  $r$  of the [CLS] token in [CLS]  $C_A X_A$  [SEP]  $C_B X_B$  [SEP] ( $r \in$  Random, Linked, Grouped). The rationale behind this is to encourage the model to learn the relevance between concepts in the ESCO taxonomy. We formalize the objectives in [Equation \(8.1\)](#):

<sup>5</sup>The special tokens used in this example follow the naming convention of BERT for readability, [CLS] and [SEP]. However, since we use XLM-R<sub>large</sub> there are different special tokens: <s> as the beginning of the sequence, </s> as the SEP token, and </s></s> as segment separators. Formally, given the example in the text: <s>  $C_A X_A$  </s></s>  $C_B X_B$  </s>.

Dataset Name	Lang.	Loc.	License	Task	Metric	Input Type	Train	Dev.	Test
SKILLSPAN	en	*	CC-BY-4.0	SL	Span-F1	Sentences	5,866	3,992	4,680
SAYFULLINA	en	UK	Unknown	SL	Span-F1	Sentences	3,706	1,854	1,853
GREEN	en	UK	CC-BY-4.0	SL	Span-F1	Sentences	8,670	963	336
JOBSTACK	en	*	RLT	SL	Span-F1	Sentences	18,055	2,082	2,092
BHOLA	en	SG	CC-BY-4.0	MLC	MRR	Documents	16,238	2,030	2,030
KOMPETENCER	en	DK	CC-BY-4.0	MCC	W. Macro-F1	Skills	9,472	1,577	1,578
KOMPETENCER	da	DK	CC-BY-4.0	MCC	W. Macro-F1	Skills	138	-	784
GNEHM	de	CH	CC-BY-NC-SA-4.0	SL	Span-F1	Sentences	22,134	2,679	2,943
FIJO	fr	FR	Unknown	SL	Span-F1	Sentences	399	50	50

**Table 8.1: Dataset Statistics.** We show statistics for all 9 JAD datasets. There are 6 datasets in English and 3 in other languages (Danish, German, and French). We indicate the location the JAD originates from (whenever applicable, \* indicates it comes from a variety of countries). We indicate the license of the dataset. Most of the task types consist of sequence labeling (e.g., span extraction, Named Entity Recognition, soft skill tagging). To maintain consistency, we use a single metric for each task type: Sequence Labeling (SL), Multilabel Classification (MLC), and Multiclass Classification (MCC). For KOMPETENCER, the statistics are provided in brackets for the Danish language.

$$\begin{aligned}
 \mathcal{L} &= \mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{ERP}} \\
 &= - \sum_i \log p(x_i | \mathbf{h}_i) - \log p(r | \mathbf{h}_{[\text{CLS}]}) ,
 \end{aligned} \tag{8.1}$$

we define the overall loss  $\mathcal{L}$  as the sum of the MLM loss  $\mathcal{L}_{\text{MLM}}$  and the ERP loss  $\mathcal{L}_{\text{ERP}}$ . The MLM loss is calculated as the negative log probability of the input token  $x_i$  given the representation  $\mathbf{h}_i$ . Similarly, the ERP loss is the negative log probability of the relationship  $r$  given the representation of the start-token  $\mathbf{h}_{[\text{CLS}]}$ . In our implementation, we use XLM-R<sub>large</sub> and classify the start-token [CLS] for ERP to improve the model’s ability to capture the relationships between ESCO occupations and skills.

### 8.2.6 Implementation

For optimization we follow (Yasunaga et al., 2022a), we use the AdamW (Loshchilov and Hutter, 2019) optimizer with  $(\beta_1, \beta_2) = (0.9, 0.98)$ . We warm up the learning rate  $1e^{-5}$  for a ratio of 6% and then

linearly decay it. The model is trained for 30K steps, which is equivalent to one epoch over the data, and the training process takes 33 hours on one A100 GPU with tf32. We use a development set comprising 1% of the data for evaluation. In [Figure 8.3](#), the pre-training loss and performance on the dev. set are plotted, it can be seen that the accuracy plateaus at 30K steps. Though the train and development loss hint that further gains could be obtained on the pretraining objective, we found through empirical analysis on downstream tasks that 30K steps performs best.

## 8.3 Experimental Setup

[Table 8.1](#) provides the details of the downstream datasets used in this study. Most of the datasets are in EN, with a smaller number in DA, DE, and FR. For each dataset, a brief description and the corresponding best-performing models are given. We put examples of each dataset (apart from JobStack due to the license) in [Subsection 8.8.2](#).

### 8.3.1 SKILLSPAN ([Zhang et al., 2022a](#))

The job posting dataset includes annotations for skills and knowledge, derived from the ESCO taxonomy. The best model in the relevant paper, JobBERT, was retrained using a DAPT approach on a dataset of 3.2 million EN job posting sentences. This is the best-performing model which we will compare against.

### 8.3.2 KOMPETENCER ([Zhang et al., 2022c](#))

This dataset is used to evaluate models on the task of classifying skills according to their ESCO taxonomy code. It includes EN and DA splits, with the EN set derived from SKILLSPAN. There are three experimental setups for evaluation: fully supervised with EN data, zero-shot classification (EN→DA), and few-shot classification (a few DA instances). The best-performing model in this work is RemBERT ([Chung et al., 2020](#)), which

obtains the highest weighted macro-F1 for both EN and DA. In this work, we use setup 1 and 3, where all available data is used.

### 8.3.3 BHOLA (Bhola et al., 2020)

The task of this EN job posting dataset is multilabel classification: Predicting a list of necessary skills in for a given job description. It was collected from a Singaporean government website. It includes job requirements and responsibilities as data fields. Pre-processing steps included lowercasing, stopword removal, and rare word removal. Their model is BERT with a bottleneck layer (Liu et al., 2017a). In our work, the bottleneck layer is not used and no additional training data is generated through bootstrapping. To keep comparison fair, we re-train their model without the additional layer and bootstrapping. We use Mean Reciprocal Rank as the main results metric.

### 8.3.4 SAYFULLINA (Sayfullina et al., 2018)

This dataset is used for soft skill prediction, a sequence labeling problem. Soft skills are personal qualities that contribute to success, such as “team working”, “being dynamic”, and “independent”. The models for this dataset include a CNN (Kim, 2014), an LSTM (Hochreiter et al., 1997), and a Hierarchical Attention Network (Yang et al., 2016). We compare to their best-performing LSTM model.

### 8.3.5 GREEN (Green et al., 2022)

A sentence-level sequence labeling task involving labeling skills, qualifications, job domain, experience, and occupation labels. The job positions in the dataset are from the United Kingdom. The industries represented in the data vary and include IT, finance, healthcare, and sales. Their model for this task is a Conditional Random Field (Lafferty et al., 2001) model.



### 8.3.6 JOBSTACK (Jensen et al., 2021b)

This corpus is used for de-identifying personal data in job vacancies on Stack Overflow. The task involves sequence labeling and predicting Organization, Location, Name, Profession, and Contact details labels. The best-performing model for this task is a transformer-based (Vaswani et al., 2017) model trained in a multi-task learning setting. Jensen et al. (2021b) propose to use the I2B2/UTHealth corpus, which is a medical de-identification task (Stubbs and Uzuner, 2015a), as auxiliary data, which showed improvement over their baselines.

### 8.3.7 GNEHM (Gnehm et al., 2022b)

A Swiss-German job ad dataset where the task is Information and Communications Technology (ICT)-related entity recognition, these could be ICT tasks, technology stack, responsibilities, and so forth. The used dataset is a combination of two other Swiss datasets namely the Swiss Job Market Monitor and an online job ad dataset (Gnehm and Clematide, 2020; Buchmann et al., 2022). Their model is dubbed JobGBERT and is based on DAPT with German BERT<sub>base</sub> (Chan et al., 2020).

### 8.3.8 FIJO (Beauchemin et al., 2022)

A French job ad dataset with the task of labeling skill types using a sequence labeling approach. The skill groups are based on the AQESSS public skills repositories and proprietary skill sets provided by their collaborators. These skill types are divided into four categories: “Thoughts”, “Results”, “Relational”, and “Personal”. The best-performing model for this task is CamemBERT (Martin et al., 2020b).

## 8.4 Results

The results of the models are presented in Table 8.2. To evaluate the performance, four different models are used in total: ESCOXLM-R, the

Dataset	Lang.	Metric	Prev. SOTA	XLM-R <sub>large</sub>	XLM-R <sub>large</sub> (+ DAPT)	ESCOXML-R	$\Delta$
SKILLSPAN	EN	Span-F1	58.9 $\pm$ 4.5	59.7 $\pm$ 4.6	62.0 $\pm$ 4.0	<b>62.6</b> $\pm$ 3.7	+3.7
SAYFULLINA	EN	Span-F1	73.1 $\pm$ 2.1	89.9 $\pm$ 0.5	90.6 $\pm$ 0.4	<b>92.2</b> $\pm$ 0.2	+19.1
GREEN	EN	Span-F1	31.8 $\pm$ *	49.0 $\pm$ 2.4	47.5 $\pm$ 0.7	<b>51.2</b> $\pm$ 2.1	+19.4
JOBSTACK	EN	Span-F1	<b>82.1</b> $\pm$ 0.8	81.2 $\pm$ 0.6	80.4 $\pm$ 0.7	82.0 $\pm$ 0.7	-0.1
KOMPETENCER	EN	W. Macro-F1	62.8 $\pm$ 2.8	59.0 $\pm$ 9.5	<b>64.3</b> $\pm$ 0.5	63.5 $\pm$ 1.3	-0.7
BHOLA	EN	MRR	90.2 $\pm$ 0.2	90.5 $\pm$ 0.3	90.0 $\pm$ 0.3	<b>90.7</b> $\pm$ 0.2	+0.5
GNEHM	DE	Span-F1	86.7 $\pm$ 0.4	87.1 $\pm$ 0.4	86.8 $\pm$ 0.2	<b>88.4</b> $\pm$ 0.5	+1.7
FIJO	FR	Span-F1	31.7 $\pm$ 2.3	41.8 $\pm$ 2.0	41.7 $\pm$ 0.7	<b>42.0</b> $\pm$ 2.3	+10.3
KOMPETENCER	DA	W. Macro-F1	45.3 $\pm$ 1.5	41.2 $\pm$ 9.8	<b>45.6</b> $\pm$ 0.8	45.0 $\pm$ 1.4	-0.3

Table 8.2: **Results of Experiments.** The datasets and models are described in Section 8.3. We re-train the best-performing models of all papers to give us the standard deviation. The best-performing model is in bold. The difference in performance between ESCOXML-R and the previous SOTA is shown as  $\Delta$ . Note (\*) that the results for GREEN are based on a CRF model where the data has been pre-split, and therefore, there is no standard deviation.

best-performing model originally reported in the relevant paper for the downstream task, vanilla XLM-R<sub>large</sub>, and an XLM-R<sub>large</sub> model that we continuously pre-trained using only MLM (DAPT; excluding the ERP objective) using the same pre-training hyperparameters as ESCOXML-R. For more information regarding the hyperparameters of fine-tuning, we refer to Subsection 8.8.3 (Table 8.5).

### 8.4.1 English

ESCOXML-R is the best-performing model in 4 out of 6 EN datasets. The largest improvement compared to the previous SOTA is observed in SAYFULLINA and GREEN, with over 19 F1 points. In 3 out of 4 datasets, ESCOXML-R has the overall lower standard deviation. For JOBSTACK, the previous SOTA performs best, and for KOMPETENCER, XLM-R<sub>large</sub> (+ DAPT) has the highest performance.

### 8.4.2 Non-English

In 2 out of 3 datasets, ESCOXML-R improves over the previous SOTA, with the largest absolute difference on French FIJO with 10.3 F1 points. In the

Danish subset of KOMPETENCER, XLM-R<sub>large</sub> (+ DAPT) has higher performance than ESCOXML-R. Next, we will discuss potential reasons for these differences.

### 8.4.3 Analysis

We highlight that the performance gains of ESCOXML-R are generally much larger than any of the losses, indicating a largely positive effect of training on ESCO. The improved performance of ESCOXML-R on JAD datasets in [Table 8.2](#) is likely due to the focus on tasks with token-level annotation (i.e., sequence labeling). This suggests that pre-training on the ESCO taxonomy is particularly useful for these types of tasks. The under-performance of ESCOXML-R on the KOMPETENCER dataset in both EN and DA may be because the task involves predicting the ESCO taxonomy code for a given skill *without context*, where we expect ESCO to particularly help with tasks where having context is relevant. We suspect applying DAPT and ERP on ESCO specifically improves recognizing entities that are uncommon. On the other hand, the poor performance on the JOBSTACK dataset may be due to the task of predicting various named entities, such as organizations and locations. By manual inspection, we found that ESCO does not contain entities related to organizations, locations, or persons, thus this reveals that there is a lack of relevant pre-training information to JOBSTACK.

## 8.5 Discussion

### 8.5.1 Performance on Span Length

We seek to determine whether the difference in performance between the ESCOXML-R and XLM-R<sub>large</sub> models is due to shorter spans, and to what extent. One application of predicting short spans well is the rise of technologies, for which the names are usually short in length. [Zhang et al. \(2022d\)](#) observes that skills described in the ESCO dataset are typically short, with a median length of approximately 3 tokens. We compare the average performance of both models on the test sets of each dataset, where

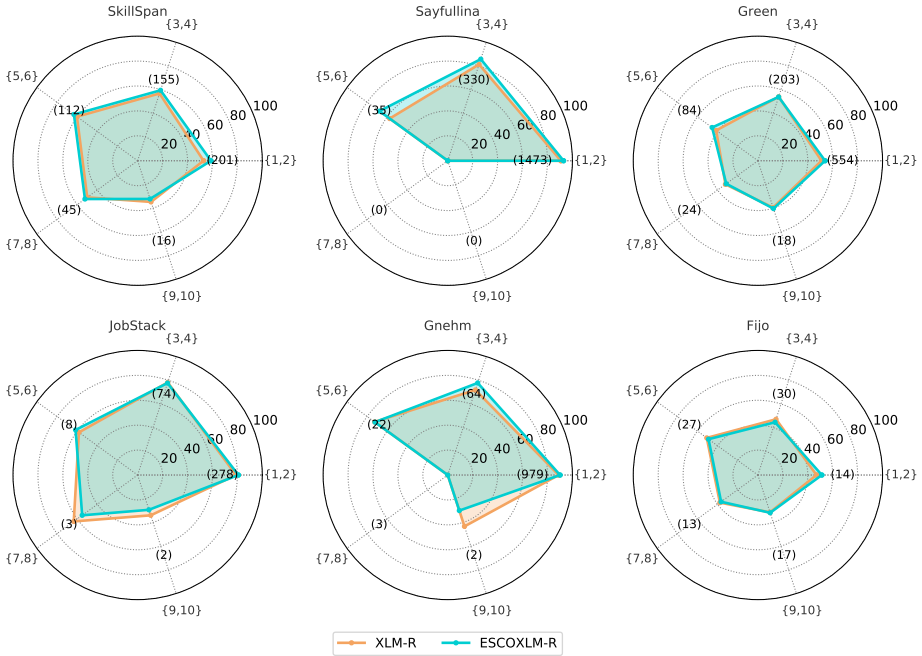


Figure 8.4: **Radar Charts of Span-F1 performance by Span Token Length.** We show the performance of  $\text{XLM-R}_{\text{large}}$  and ESCOXML-R on different span lengths, we bucketed the performances of both models according to the length of the spans, up to 10 tokens, and presented the average performance over five random seeds. We did not include error bars in these plots. Note that in some plots, there are no instances in certain buckets (e.g., SAYFULLINA with 7-8, 9-10). Also, some outer rings only go up to 60 span F1, rather than 100.

span-F1 is used as measurement. We group gold spans into buckets of lengths 1-2, 3-4, 5-6, 7-8, and 9-10, and present the span-F1 for each model ( $\text{XLM-R}_{\text{large}}$  vs. ESCOXML-R) in each bucket.

Shown in Figure 8.4, ESCOXML-R outperforms  $\text{XLM-R}_{\text{large}}$  on shorter spans (i.e., 1-2 or 3-4) in 6 out of the 6 datasets, suggesting that pre-training on ESCO is beneficial for predicting short spans. However, there is a slight decline in performance on some datasets (e.g., SKILLSPAN, JOBSTACK, and GNEHM) when the spans are longer (i.e., 7-8 or 9-10). It is

Dataset	Ratio	Span-F1 (Entity)		Span-F1 (Surface)	
		XML-R	ESCOXML-R	XML-R	ESCOXML-R
SKILLSPAN	0.90	59.9 $\pm$ 7.9	<b>61.6</b> $\pm$ 6.6	56.4 $\pm$ 5.7	<b>57.9</b> $\pm$ 4.3
SAYFULLINA	0.22	94.0 $\pm$ 0.2	<b>95.7</b> $\pm$ 0.3	82.8 $\pm$ 0.6	<b>87.2</b> $\pm$ 0.7
GREEN	0.79	50.3 $\pm$ 2.4	<b>53.1</b> $\pm$ 2.1	49.2 $\pm$ 2.4	<b>52.0</b> $\pm$ 2.1
JOBSTACK	0.41	85.6 $\pm$ 0.7	<b>86.4</b> $\pm$ 0.5	78.4 $\pm$ 1.2	<b>79.8</b> $\pm$ 0.7
GNEHM	0.53	89.3 $\pm$ 0.3	<b>89.6</b> $\pm$ 0.4	87.3 $\pm$ 0.3	<b>87.8</b> $\pm$ 0.6
FIJO	0.77	34.4 $\pm$ 2.9	<b>35.7</b> $\pm$ 1.1	34.4 $\pm$ 1.1	<b>35.7</b> $\pm$ 1.1

Table 8.3: **Entity vs. Surface-level span-F1 on Test.** In this table, the performance of two systems, XML-R<sub>large</sub> and ESCOXML-R, was measured using entity-level and surface-level span-F1 scores. Entity-level span-F1 measures precision, recall, and harmonic mean at the entity level, while surface-level span-F1 measures a system’s ability to recognize a range of entities. We include the ratio of surface entities to total entities in each *training* set, with a higher ratio indicating more variety (a ratio of 1.00 indicates all entities are unique).

worth noting that the number of instances in these longer span buckets is lower, and therefore errors may be less apparent in terms of their impact on overall performance.

### 8.5.2 Entity-F1 vs. Surface-F1

In this analysis, we adopt the evaluation method used in the W-NUT shared task on Novel and Emerging Entity Recognition (Derczynski et al., 2017). In this shared task, systems are evaluated using two measures: entity span-F1 and surface span-F1. Entity span-F1 assesses the precision, recall, and harmonic mean (F1) of the systems at the entity level, while surface span-F1 assesses their ability to correctly recognize a diverse range of entities, rather than just the most frequent surface forms. This means surface span-F1 counts entity types, in contrast to entity tokens in the standard entity span-F1 metric.

As shown in Table 8.3, we first calculate the ratio of unique entities and total entities in each relevant train set (i.e., datasets where we do span

labeling). A higher ratio number indicates a wider variety of spans. Both XLM- $R_{\text{large}}$  and ESCOXML-R tend to have lower performance when variety gets high (above 0.75). In addition, there are 2 datasets (SAYFULLINA, JOBSTACK) where we see a low variety of spans and large discrepancy between performance of entity span-F1 and surface span-F1. This difference is lower for ESCOXML-R (especially in SAYFULLINA) suggesting that pre-training on ESCO helps predicting uncommon entities.

It is also noteworthy that the standard deviations for the scores at the entity span-F1 are generally lower than those for the surface span-F1. This suggests that the results for the entity span-F1 scores are more consistent across different runs, likely due to recognizing common entities more.

Overall, ESCOXML-R consistently outperforms XLM- $R_{\text{large}}$  in both the entity-level and surface-level F1 scores, indicating the benefits of using the ESCO dataset for pre-training on JAD tasks.

## 8.6 Related Work

To the best of our knowledge, we are the first to internalize an LM with ESCO for job-related NLP tasks. There are, however, several works that integrate factual knowledge (i.e., knowledge graphs/bases) into an LM. [Peters et al. \(2019\)](#) integrates multiple knowledge bases into LMs to enhance their representations with structured, human-curated knowledge and improve perplexity, fact recall and downstream performance on various tasks. [Zhang et al. \(2019\)](#); [He et al. \(2020\)](#); [Wang et al. \(2021b\)](#) combine LM training with knowledge graph embeddings. [Wang et al. \(2021a\)](#) introduces K-Adapter for injecting knowledge into pre-trained models that adds neural adapters for each kind of knowledge domain. [Yu et al. \(2022\)](#) introduces Dict-BERT, which incorporates definitions of rare or infrequent words into the input sequence and further pre-trains a BERT model.

[Calixto et al. \(2021\)](#) introduced a multilingual Wikipedia hyperlink prediction intermediate task to improve language model pre-training. Similarly, [Yasunaga et al. \(2022a\)](#) introduced LinkBERT which leverages links between documents, such as hyperlinks, to capture dependencies and

knowledge that span across documents by placing linked documents in the same context and pre-training the LM with MLM and document relation prediction.

## 8.7 Conclusion

In this study, we introduce ESCOXML-R as a multilingual, domain-adapted LM that has been further pre-trained on the ESCO taxonomy. We evaluated ESCOXML-R, to the best of our knowledge, on the broadest evaluation set in this domain on 4 different languages. The results showed that ESCOXML-R outperformed XLM-R<sub>large</sub> on job-related downstream tasks in 6 out of 9 datasets, particularly when the task was relevant to the ESCO taxonomy and context was important. It was found that the improvement of ESCOXML-R was mainly due to its performance on shorter span lengths, demonstrating the value of pre-training on the ESCO dataset. ESCOXML-R also demonstrated improved performance on both frequent surface spans and a wider range of spans. Overall, this work showed the potential of ESCOXML-R as an LM for multilingual job-related tasks. We hope that it will encourage further research in this area.

## Limitations

There are several limitations to this study that should be considered. First, a key limitation is the lack of a variety of language-specific JAD. Here, we have four different languages namely EN, DA, FR, and DE. This means that our analysis is based on a limited subset of languages and may not be representative of JAD data outside of these four languages.

In turn, the second limitation is that the ESCO taxonomy used as pre-training data only covers Europe and the datasets used in this work also covers mostly Europe. The results may not be generalizable to other regions. However, we see a slight improvement in the BHOLA dataset, the data of which comes from Singapore, which hints that it could generalize to other cultures.

The ESCO relation prediction task aims for learning the relations between elements of the ESCO taxonomy. We acknowledge that we do not evaluate the effectiveness of the pre-training objective in relation-centered tasks. Unfortunately, to the best of our knowledge, there is no job-related dataset containing relations between skill/occupation concepts to benchmark our model on. We consider this interesting future work.

Finally, we did not conduct an ablation study on the ERP pre-training objective, i.e., which errors it makes. As the accuracy of the objective is 60%, we are unable to determine which sampling method is detrimental to this accuracy. However, we suspect that the Linked sampling approach might be the hardest to predict correctly. For example, many occupations have a lot of necessary and optional skills, thus it is harder to determine if some skill truly belongs to a specific occupation. Nevertheless, we see that adding the ERP objective improves over regular MLM domain-adaptive pre-training.

Despite these limitations, we believe that this study provides valuable resources and insights into the use of ESCOXLM-R for analyzing JAD and suggests directions for future research. Future studies could address the limitations of this study by using a larger, more diverse datasets and by conducting ablation studies on the language model to better understand which parts contribute to the results.

## **Ethics Statement**

We also see a potential lack of language inclusiveness within our work, as we addressed in the Limitation section that ESCO mostly covers Europe (and the Arabic language). Nevertheless, we see ESCOXLM-R as a step towards inclusiveness, due to JAD frequently being English-only. In addition, to the best of our knowledge, ESCO itself is devoid of any gendered language, specifically, pronouns and other gender-specific terms in, e.g., occupations. However, we acknowledge that LMs such as ESCOXLM-R could potentially be exploited in the process of hiring candidates for a specific job with unintended consequences (unconscious bias and dual



use). There exists active research on fairer recommender systems (e.g., bias mitigation) for human resources (e.g., [Mujtaba and Mahapatra, 2019](#); [Raghavan et al., 2020](#); [Deshpande et al., 2020](#); [Köchling and Wehner, 2020](#); [Sánchez-Monedero et al., 2020](#); [Wilson et al., 2021](#); [van Els et al., 2022](#); [Arafan et al., 2022](#)).

## **8.8 Appendix**

### **8.8.1 Example Extraction from ESCO**

```
1 {
2   "id": int, "esco_code": "2250.4",
3   "preferred_label": "animal therapist",
4   "major_group": {
5     "title": "Veterinarians",
6     "description": "Veterinarians diagnose, [...]"
7   },
8   "alternative_label": [ "animal convalescence
9     therapist",
10    [...]
11  ],
12  "description": "Animal therapists provide [...]",
13  "essential_skills": [
14    {
15      "title": "anatomy of animals",
16      "description": "The study of animal body
17        parts,
18        [...]"
19    },
20    [...]
21  ],
22  "optional_skills": [...]
23 }
```

Listing 1: **Example Extraction.** An example of the information that is given for ESCO code 2250.4: animal therapist. The original page can be found here: <http://data.europa.eu/esco/occupation/0b2d3242-22a3-4de5-bd29-efd39cdf2c31>.

## 8.8.2 Data Examples

SKILLSPAN	<a href="#">Listing 2</a>
SAYFULLINA	<a href="#">Listing 3</a>
GREEN	<a href="#">Listing 4</a>
BHOLA	<a href="#">Listing 5</a>
KOMPETENCER	<a href="#">Listing 6</a>
FIJO	<a href="#">Listing 7</a>
GNEHM	<a href="#">Listing 8</a>

Table 8.4: **Data Examples.** Data example references for each dataset for fine-tuning.

1	Experience	0	0
2	in	0	0
3	working	B-Skill	0
4	on	I-Skill	0
5	a	I-Skill	0
6	cloud-based	I-Skill	0
7	application	I-Skill	0
8	running	0	0
9	on	0	0
10	Docker	0	B-Knowledge
11	.	0	0
12			
13	A	0	0
14	degree	0	B-Knowledge
15	in	0	I-Knowledge
16	Computer	0	I-Knowledge
17	Science	0	I-Knowledge
18	or	0	0
19	related	0	0
20	fields	0	0
21	.	0	0

Listing 2: Data Example SkillSpan.

```
1  ability      0
2  to           0
3  work        B-Skill
4  under       I-Skill
5  stress      I-Skill
6  condition   0
7
8  due         0
9  to          0
10 the         0
11 dynamic     B-Skill
12 nature      0
13 of          0
14 the         0
15 group       0
16 environment 0
17 ,           0
18 the         0
19 ideal       0
20 candidate   0
21 will        0
```

Listing 3: Data Example Sayfullina.

1	A	0
2	sound	0
3	understanding	0
4	of	0
5	the	0
6	Care	B-Skill
7	Standards	I-Skill
8	together	0
9	with	0
10	a	0
11	Nursing	B-Qualification
12	qualification	I-Qualification
13	and	0
14	current	0
15	NMC	B-Qualification
16	registration	I-Qualification
17	are	0
18	essential	0
19	for	0
20	this	0
21	role	0
22	.	0

Listing 4: Data Example Green.

1	department economics national university singapore
2	invites applications teaching oriented positions
3	level lecturer senior lecturer [...] <labels>

Listing 5: Data Example Bhola.

1	<English>	
2	team worker	S4
3	passion for developing your career	S1
4	liaise with internal teams	S1
5	identify system requirements	S2
6	plan out our new features	S4
7		
8	<Danish>	
9	arbejde med børn i alderen $\frac{1}{2}$ -3 år	S3
10	samarbejde	S1
11	fokusere på god kommunikation	S1
12	bidrage til at styrke fællesskabet	S1
13	ansvarsbevidst	A1
14	lyst til et aktivt udeliv	A1

Listing 6: Data Example Kompetencer.



1	Participer	B-relationnel
2	au	I-relationnel
3	réseau	I-relationnel
4	téléphonique	I-relationnel
5	mis	0
6	sur	0
7	pied	0
8	lors	0
9	des	0
10	campagnes	0
11	d'inscription	0
12	pour	0
13	fournir	B-pensee
14	les	I-pensee
15	renseignements	I-pensee
16	nécessaires	I-pensee
17	aux	I-pensee
18	assurés	I-pensee

Listing 7: Data Example Fijo.

1	in	0
2	mit	0
3	guten	0
4	EDV-Kenntnissen	B-ICT
5		
6	.	0
7	Es	0
8	erwartet	0
9	Sie	0
10	eine	0
11	interessante	0
12	Aufgabe	0
13	in	0
14	einer	0
15	Adressverwaltung	0
16	(	0
17	Rechenzenter	B-ICT
18	)	0

Listing 8: Data Example Gnehm.

	Learning rate	Batch size	max_seq_length	Epochs
SKILLSPAN	$\{1e^{-4}, 5e^{-5}, 1e^{-5}, 5e^{-6}\}$	{16, 32, 64}	128	20
KOMPETENCER	$\{1e^{-4}, 7e^{-5}, 5e^{-5}, 1e^{-5}, 5e^{-6}\}$	{8, 16, 32}	128	20
BHOLA	$\{1e^{-4}, 7e^{-5}, 5e^{-5}, 1e^{-5}, 5e^{-6}\}$	{4, 16, 32, 64, 128}	{128, 256}	10
SAYFULLINA	$\{1e^{-4}, 5e^{-5}, 1e^{-5}\}$	{16, 32, 64}	128	10
GREEN	$\{1e^{-4}, 5e^{-5}, 1e^{-5}\}$	{16, 32, 64}	128	10
JOBSTACK	$\{1e^{-4}, 7e^{-5}, 5e^{-5}, 1e^{-5}, 5e^{-6}\}$	{16, 32, 64, 128}	128	20
GNEHM	$\{1e^{-4}, 5e^{-5}, 1e^{-5}\}$	{16, 32, 64}	128	5
FIJO	$\{1e^{-4}, 5e^{-5}, 1e^{-5}\}$	{8, 16, 32, 64}	128	10

Table 8.5: **Hyperparameter Sweep for Fine-tuning.** We show a hyperparameter sweep for fine-tuning all models. Learning rate differs for both XLM-R<sub>large</sub> and ESCOXML-R, where XLM-R<sub>large</sub> performs best on lower learning rate (e.g.,  $1e^{-5}$ ) and ESCOXML-R on a bit of a higher learning rate (e.g.,  $5e^{-5}$ ). A batch size of 32 works best for all models. The max sequence length is usually the same, except for BHOLA due to it containing long texts. Epochs are determined based on previous work (i.e., the relevant datasets).

### 8.8.3 Fine-tuning Details

For fine-tuning XLM-R<sub>large</sub> (+ DAPT) and ESCOXML-R on the downstream tasks, we use MaChAmp (van der Goot et al., 2021b). For more details we refer to their paper. We always include the original learning rate, batch size, maximum sequence length, and epochs from the respective downstream tasks in our search space (whenever applicable). Each model is trained on an NVIDIA A100 GPU with 40GBs of VRAM and an AMD Epyc 7662 CPU. The seed numbers the models are initialized with are 276800, 381552, 497646, 624189, 884832. We run all models with the maximum number of epochs indicated in Table 8.5 and select the best-performing one based on validation set performance in the downstream metric.



# Chapter 9

## NNOSE: Nearest Neighbor Occupational Skill Extraction

The work presented in this chapter is based on a paper accepted as: Mike Zhang, Rob van der Goot, Min-Yen Kan, and Barbara Plank. NNOSE: Nearest neighbor occupational skill extraction. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 589–608, St. Julian’s, Malta, March 2024a. Association for Computational Linguistics. URL <https://aclanthology.org/2024.eacl-long.35>.

## Abstract

The labor market is changing rapidly, prompting increased interest in the automatic extraction of occupational skills from text. With the advent of English benchmark job description datasets, there is a need for systems that handle their diversity well. We tackle the complexity in occupational skill datasets tasks—combining and leveraging multiple datasets for skill extraction, to identify rarely observed skills within a dataset, and overcoming the scarcity of skills across datasets. In particular, we investigate the retrieval-augmentation of language models, employing an external datastore for retrieving similar skills in a dataset-unifying manner. Our proposed method, **N**earest **N**eighbor **O**ccupational **S**kill **E**xtraction (NNOSE) effectively leverages multiple datasets by retrieving neighboring skills from other datasets in the datastore. This improves skill extraction *without* additional fine-tuning. Crucially, we observe a performance gain in predicting infrequent patterns, with substantial gains of up to 30% span-F1 in cross-dataset settings.

## 9.1 Introduction

Labor market dynamics, influenced by technological changes, migration, and digitization, have led to the availability of job descriptions (JD) on platforms to attract qualified candidates (Brynjolfsson and McAfee, 2011, 2014; Balog et al., 2012). JDs consist of a collection of skills that exhibit a characteristic *long-tail pattern*, where popular skills are more common while niche expertise appears less frequently across industries (Autor et al., 2003; Autor and Dorn, 2013), such as “teamwork” vs. “system design”.<sup>1</sup> This pattern poses challenges for skill extraction (SE) and analysis, as certain skills may be underrepresented, overlooked, or emerging in JDs. This complexity makes the extraction and analysis of skills more difficult, resulting in a *sparsity of skills* in SE datasets. We tackle this by combining

---

<sup>1</sup>Examples are from the [CEDEFOP Skill Platform](#).

three different skill datasets.

To address the challenges in SE, we explore the potential of Nearest Neighbors Language Models (NNLMs; [Khandelwal et al., 2020](#)). NNLMs calculate the probability of the next token by combining a parametric language model (LM) with a distribution derived from the  $k$ -nearest context-token pairs in the datastore. This enables the storage of large amounts of training instances without the need to retrain the LM weights, improving language modeling. However, the extent to which NNLMs enhance application-specific end-task performance beyond language modeling remains relatively unexplored. Notably, NNLMs offer several advantages, as highlighted by [Khandelwal et al. \(2020\)](#): First, explicit memorization of the training data aids generalization. Second, a single LM can adapt to multiple domains without domain-specific training, by incorporating domain-specific data into the datastore (e.g., multiple datasets). Third, the NNLM architecture excels at predicting rare patterns, particularly the long-tail.

Therefore, we seek to answer the question: *How effective are nearest neighbors retrieval methods for occupational skill extraction?* Our contributions are as follows:

- To the best of our knowledge, we are the first to investigate encoder-based  $k$ NN retrieval by leveraging *multiple* datasets.
- Furthermore, we present a novel domain-specific RoBERTa<sub>base</sub>-based language model, JobBERTa, tailored to the job market domain.
- We conduct an extensive analysis to show the advantages of  $k$ NN retrieval, in contrast to prior work that primarily focuses on hyperparameter-specific analysis.<sup>2</sup>

---

<sup>2</sup>Code: <https://github.com/jjzha/nnose>.

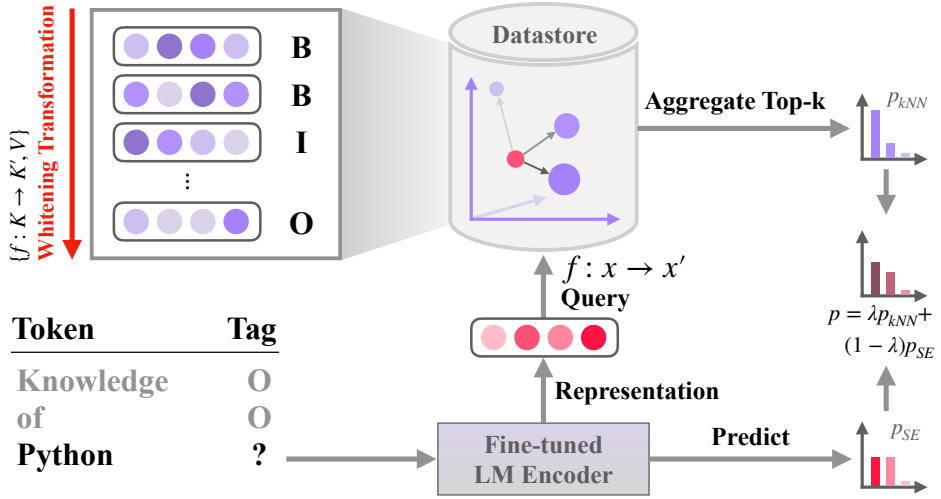


Figure 9.1: **Setup of NNOSE.** The datastore consists of paired contextual token representations obtained from a fine-tuned encoder and the corresponding BIO tag. We use a whitening transformation to enhance the isotropy of token representations. During inference, i.e., retrieving tokens, we use the same whitening transformation on the test token’s representation to retrieve the  $k$ -nearest neighbors from the datastore. We interpolate the encoder and  $k$ NN distributions with a hyperparameter  $\lambda$  as the final distribution.

## 9.2 Nearest Neighbor Skill Extraction

### 9.2.1 Skill Extraction.

The task of SE is formulated as a sequence labeling problem. We define a set of job description sentences  $\mathcal{X}$ , where each  $d \in \mathcal{X}$  represents a set of sequences with the  $j^{\text{th}}$  input sequence  $\mathcal{X}_d^j = \{x_1, x_2, \dots, x_i\}$ , with a corresponding target sequence of BIO-labels  $\mathcal{Y}_d^j = \{y_1, y_2, \dots, y_i\}$ . The labels include “B” (beginning of a skill token), “I” (inside skill token), and “O” (any outside token). The objective is to use  $\mathcal{D}$  in training a labeling algorithm that accurately predicts entity spans by assigning an output label  $y_i$  to each token  $x_i$ .



## 9.2.2 NNOSE

The core idea of NNOSE is that we augment the extraction of skills during inference with a  $k$ NN retrieval component and a datastore consisting of context–token pairs. Figure 9.1 outlines our two-step approach. First, we extract skills by getting token representation  $\mathbf{h}_i$  from  $x_i$  and assign a probability distribution  $p_{\text{SE}}$  for each  $\mathbf{h}_i$  in the input sentence. Second, we use each  $\mathbf{h}_i$  to find the most similar token representations in the datastore and get the probability distribution  $p_{\text{kNN}}$ , aggregated from the  $k$ -nearest context–token pairs. Last, we obtain the final probability distribution  $p$  by interpolating between the two distributions. In addition to formalizing NNOSE, we apply the Whitening Transformation (Subsection 9.2.5) to the embeddings, an important process for  $k$ NN approaches as used in previous work (Su et al., 2021; Yin and Shang, 2022).

## 9.2.3 Datastore

The datastore  $\mathcal{D}$  comprises key–value pairs  $(\mathbf{h}_i, y_i)$ , where each  $\mathbf{h}_i$  represents the contextualized token embedding computed by a *fine-tuned* SE encoder, and  $y_i \in \{\text{B}, \text{I}, \text{O}\}$  denotes the corresponding gold label. Typically, the datastore consists of all tokens from the training set. In contrast to the approach employed by Wang et al. (2022b) for  $k$ NN–NER, where they only store B and I tags in the datastore (only named entities), we also include the O-tag in the datastore. This allows us to retrieve non-named entities, which is more intuitive than assigning non-entity probability mass to the B and I tokens.

## 9.2.4 Inference

During inference, the NNOSE model aims to predict  $y_i$  based on the contextual representation of  $x_i$  (i.e.,  $\mathbf{h}_i$ ). This representation is used to query the datastore for  $k$ NN using an  $L^2$  distance measure (following Khandelwal et al., 2020), denoted as  $d(\cdot, \cdot)$ . Once the neighbors are retrieved, the model computes a distribution over the neighbors by applying a softmax

function with a temperature parameter  $T$  to their negative distances (i.e., similarities). This aggregation of probability mass for each label ( $B$ ,  $I$ ,  $O$ ) across all occurrences in the retrieved targets is represented as:

$$p_{\text{kNN}}(y_i | x_i) \propto \sum_{(k_i, v_i) \in \mathcal{D}} \mathbb{1}_{y=v_i} \exp\left(\frac{-d(\mathbf{h}_i, \mathbf{k})}{T}\right). \quad (9.1)$$

Items that do not appear in the retrieved targets have zero probability. Finally, we interpolate the nearest neighbors distribution  $p_{\text{kNN}}$  with the fine-tuned model distribution  $p_{\text{SE}}$  using a tuned parameter  $\lambda$  to produce the final NNOSE distribution  $p$ :

$$p(y_i | x_i) = \lambda \times p_{\text{kNN}}(y_i | x_i) + (1 - \lambda) \times p_{\text{SE}}(y_i | x_i). \quad (9.2)$$

### 9.2.5 Whitening Transformation

Several works (Li et al., 2020b; Su et al., 2021; Huang et al., 2021) note that if a set of vectors are isotropic, we can assume it is derived from the Standard Orthogonal Basis, which also indicates that we can properly calculate the similarity between embeddings. Otherwise, if it is anisotropic, we need to transform the original sentence embedding to enforce isotrophism, and then measure similarity. Su et al. (2021); Huang et al. (2021) applies the vector whitening approach (Koivunen and Kostinski, 1999) on BERT (Devlin et al., 2019b). The Whitening Transformation (WT), initially employed in data preprocessing, aims to eliminate correlations among the input data features for a model. In turn, this can improve the performance of certain models that rely on uncorrelated features. Other works (Gao et al., 2019; Ethayarajh, 2019; Li et al., 2020c; Yan et al., 2021; Jiang et al., 2022b, among others) found that (frequency) biased *token* embeddings hurt final sentence representations. These works often link token embedding bias to the token embedding anisotropy and argue it is the main reason for the bias. We apply WT to the token embeddings like previous work for nearest neighbor retrieval (Yin and Shang, 2022). In short, WT transforms the mean value of the embeddings into 0 and the covariance

Dataset	Loc.	License	Train	Dev.	Test	$\mathcal{D}$ (tokens)
SKILLSPAN	*	CC-BY-4.0	5,866	3,992	4,680	86.5K
SAYFULLINA	UK	Unknown	3,706	1,854	1,853	53.1K
GREEN	UK	CC-BY-4.0	8,670	963	336	209.5K
TOTAL						349.2K

Table 9.1: **Dataset Statistics.** We provide statistics for all three datasets, including the location and license. Input granularity is at the token level, with performance measured in span-F1. The size of the datastore  $\mathcal{D}$  is in tokens and determined by embedding tokens and their context from the training sets, resulting in approximately 350K keys. See [Subsection 9.8.2](#) for examples.

matrix into the identity matrix, and these transformations are then applied to the original embeddings. We apply WT to the embeddings before putting them in the datastore and before querying the datastore. The workflow of WT is detailed in [Subsection 9.8.1](#).

## 9.3 Experimental Setup

### 9.3.1 Data

All datasets are in English and have different label spaces. We transform all skills to the same label space and give each token a generic tag (i.e., B, I, O). We give a brief description of each dataset below and [Table 9.1](#) summarizes them:

**SKILLSPAN (Zhang et al., 2022a).** This job posting dataset includes annotations for skills and knowledge derived from the ESCO taxonomy. To fit our approach, we flatten the two label layers into one layer (i.e., BIO). The baseline is the JobBERT model, which was continuously pre-trained on a dataset of 3.2 million job posting sentences. The industries represented in the data range from tech to more labor-intensive sectors.

**SAYFULLINA (Sayfullina et al., 2018)** is used for soft skill sequence labeling. Soft skills are personal qualities that contribute to success, such as teamwork, dynamism, and independence. Data originated from the UK. This is the smallest dataset among the three, with no specified industries.

**GREEN (Green et al., 2022)**. A dataset for extracting skills, qualifications, job domain, experience, and occupation labels. The dataset consists of jobs from the UK, and the industries represented include IT, finance, healthcare, and sales. This is the largest dataset among the three.

### 9.3.2 Models

We use 3 English-based LMs: 1 general-purpose and 2 domain-specific models. Implementation details for fine-tuning and NNOSE are in [Subsection 9.8.3](#).

**JobBERT (Zhang et al., 2022a)** is a 110M parameter BERT-based model continuously pre-trained (Gururangan et al., 2020b) on 3.2M English job posting sentences. It outperforms BERT<sub>base</sub> on several skill-specific tasks.

**RoBERTa (Liu et al., 2019)**. We also use RoBERTa<sub>base</sub> (123M parameters). It showed to outperform JobBERT in our initial experiments and we therefore include this model as a baseline.

**JobBERTa (Ours)**. Given that RoBERTa outperformed JobBERT, we create another baseline and release a model named JobBERTa. This is a RoBERTa<sub>base</sub> model continuously pre-trained (Gururangan et al., 2020b) on the same 3.2M JD sentences as JobBERT.

## 9.4 Results

We evaluate the performance of fine-tuning models enhanced with NNOSE. We consider different setups: First, we compare using the Whitening

	Setting	SKILLSPAN	SAYFULLINA	GREEN	avg. span-F1
JobBERT (Zhang et al., 2022a)		60.47	88.16	42.55	63.73
+ $k$ NN	{D}+WT	61.06 $\uparrow$ 0.59	88.25 $\uparrow$ 0.09	43.56 $\uparrow$ 1.01	64.29 $\uparrow$ 0.56
+ $k$ NN	$\forall$ D+WT	60.93 $\uparrow$ 0.48	88.26 $\uparrow$ 0.10	44.44 $\uparrow$ 1.89	64.54 $\uparrow$ 0.81
RoBERTa (Liu et al., 2019)		63.88	91.97	44.49	66.78
+ $k$ NN	{D}+WT	63.57 $\downarrow$ 0.31	91.97 $-$ 0.00	45.02 $\uparrow$ 0.53	66.85 $\uparrow$ 0.07
+ $k$ NN	$\forall$ D+WT	63.98 $\uparrow$ 0.10	91.97 $-$ 0.00	44.86 $\uparrow$ 0.37	66.94 $\uparrow$ 0.16
JobBERTa (This work)		63.74	92.06	49.61	68.47
+ $k$ NN	{D}+WT	64.14 $\uparrow$ 0.40	91.89 $\downarrow$ 0.17	50.35 $\uparrow$ 0.74	68.79 $\uparrow$ 0.32
+ $k$ NN	$\forall$ D+WT	<b>64.24</b> $\uparrow$ 0.50 <sup>†</sup>	<b>92.15</b> $\uparrow$ 0.09	<b>50.78</b> $\uparrow$ 1.17 <sup>†</sup>	<b>69.06</b> $\uparrow$ 0.59

Table 9.2: **Test Set Results.** Two settings are considered for each model based on dev. set results in Subsection 9.8.8: {D} refers to the in-dataset datastore, containing keys from the specific training data, while  $\forall$ D represents a datastore with keys from all available training sets. The notation +WT indicates the application of Whitening Transformation to the keys before adding them to and querying the datastore. The performance impact of using  $k$ NN is indicated as  $\uparrow$  (increase),  $\downarrow$  (decrease), or  $-$  (no change). The best-performing setup for each dataset is highlighted. For the top-performing model (JobBERTa), <sup>†</sup> signifies statistical significance over the baseline using a token-level McNemar test (McNemar, 1947). The avg. span-F1 performance of each model across the three datasets is displayed.

Transformation (+WT) or without. Second, we explore two datastore setups: One using an in-dataset datastore ({D}), where each respective training set is stored separately, and another where all datasets are stored in the datastore ( $\forall$ D). In the latter setup, we encode all three datasets with each fine-tuned model, and each model has its own WT matrix. For example, we fine-tune a model on SKILLSPAN and encode the training set tokens of SKILLSPAN, SAYFULLINA, and GREEN to populate the datastore. From the results on the development set (Table 9.11, Subsection 9.8.8), we observe that adding WT consistently improves performance. Therefore, we only report the span-F1 scores on each test set (Table 9.2) *with* WT and the average over all three datasets.

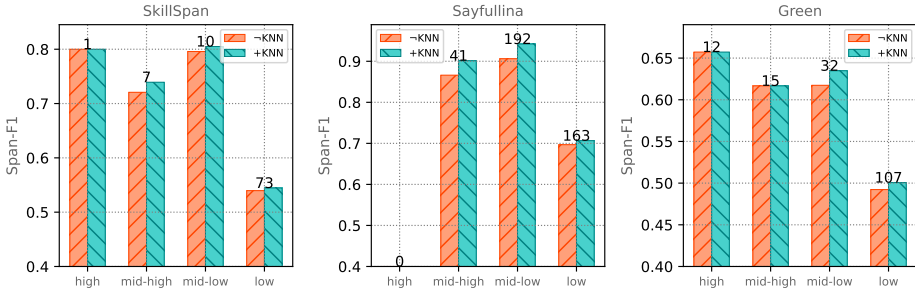


Figure 9.2: **Long-tail Prediction Performance.**  $k$ NN is based on the datastore with all the datasets. We categorize the occurrences of a skill in the test set with respect to the training set. For example, a skill in the test set occurs two times in the training set, we put this in the “low” bin. There are three frequency ranges: *high*: 10–15, *mid-high*: 7–10, *mid-low*: 4–6, *low*: 0–3. SAYFULLINA does not have any test set skills that occur more than 10 times in the training set. On top of the bars is the number of predicted skills for the test set in each bucket.

### 9.4.1 Best Model Performance

In Table 9.2, we show that the best-performing baseline model is JobBERTa, achieving more than 4 points span-F1 improvement over JobBERT and 2 points higher than RoBERTa on average. This confirms the effectiveness of DAPT in improving language models (Han and Eisenstein, 2019; Alsentzer et al., 2019b; Gururangan et al., 2020b; Lee et al., 2020a; Nguyen et al., 2020b; Zhang et al., 2022a).

### 9.4.2 Best NNOSE Setting

We confirm the trends from dev. on test: The largest improvements come from using the setup with WT, especially in the  $\forall D+WT$  setting. All models seem to benefit from the NNOSE setup, JobBERT and JobBERTa shows the largest improvements, with the largest gains observed in the  $\forall D+WT$  datastore setup. In summary,  $\forall D+WT$  consistently demonstrates performance enhancements across all experimental setups.

## 9.5 Analysis

As we store training tokens from all datasets in the datastore, we expect the model to recall a greater number of skills based on the current context during inference. In turn, this would lead to improved downstream model performance. We want to address the challenges of SE datasets by predicting long-tail patterns, and if we observe improvements in detecting unseen skills in a cross-dataset setting.

To investigate in which situations our model improves, we are analyzing the following: ① The predictive capability of NNOSE in relation to rarely occurring skills compared to regular fine-tuning (Subsection 9.5.1). Skills exhibit varying frequencies across datasets, we categorize the skill frequencies into buckets and compare the performance between vanilla fine-tuning and the inclusion of  $k$ NN. ② If NNOSE actually retrieves from other datasets when they are combined (Subsection 9.5.2), and if there is a sign of leveraging multiple datasets, then; ③ How much does NNOSE enhance performance in a cross-dataset setting (Subsection 9.5.3)? Our results indicate a large performance drop when a fine-tuned SE model, trained on one dataset, is applied to another dataset, highlighting the sparsity across datasets. We demonstrate that NNOSE helps alleviate this, both from an empirical perspective and by inspecting the prediction errors (Subsection 9.5.5).

### 9.5.1 Long-tail Skills Prediction

Khandelwal et al. (2020) observed that due to explicitly memorizing the training data, NNLMs effectively predict rare patterns. We analyze whether the performance of “long-tail skills” improves using NNOSE. A visualization of the long-tail distribution of skills is in Figure 9.8 (Subsection 9.8.9).

We present the results in Figure 9.2. We investigate the performance of JobBERTa with and without  $k$ NN based on the occurrences of skills in the evaluation set relative to the train set. We count the skills in the evaluation set that occur a number of times in the training set, ranging

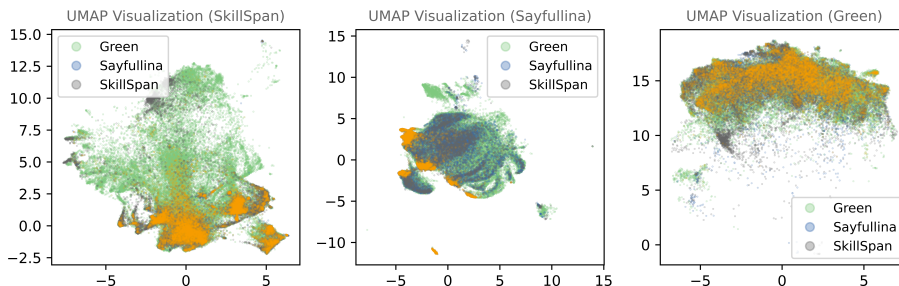
from 0–15 occurrences and is grouped into low, mid–low, mid–high, and high–frequency bins (0–3, 4–6, 7–10, 10–15, respectively). This approach estimates the number of skills the LM recalls from the training stage.

Our findings reveal that skills with low-frequency skills are the most difficult and make up the largest bucket, and our approach is able to improve on them on all three datasets. For SKILLSPAN, we observe an improvement in the low-frequency bin, from 53.9→54.5 span-F1. Similarly, GREEN exhibits a similar trend with an improvement in the low-frequency bin (49.2→50.1). Interestingly, it also shows gains in most other frequency bins. Last, for SAYFULLINA, there is also an improvement (69.7→70.7 in the low bin). It is worth pointing out that there are many skills that fall in the low bin in SKILLSPAN and GREEN. This is exactly where NNOSE improves most for these datasets. For SAYFULLINA, we notice the largest number of predicted skills is in the mid–low bin. This is where we also see improvements for NNOSE.

### 9.5.2 Retrieving From All Datasets

We presented the best improvements of NNOSE in the  $\forall D+WT$  datastore in [Section 9.4](#). An important question remains: Does the  $\forall D+WT$  setting retrieve from all datasets? Qualitatively, [Figure 9.3](#) shows the UMAP visualization ([McInnes et al., 2018](#)) of representations stored in each  $\forall D+WT$  datastore. We mark the retrieved neighbors with orange for each downstream dev. set. In all plots, we observe that GREEN is prominent in the representation space (green), while SKILLSPAN (darkcyan) and SAYFULLINA (blue) form distinct clusters. Each plot has its own pattern: SKILLSPAN and SAYFULLINA have well-shaped clusters, while GREEN consists of one large cluster. SKILLSPAN and SAYFULLINA mostly retrieve from their own clusters. In contrast, GREEN retrieves from the entire representation space, which could explain the largest span-F1 performance gains ([Table 9.2](#)). This suggests that  $k$ NN effectively leverages multiple datasets in most cases (qualitative analysis see [Subsection 9.8.10](#)).





**Figure 9.3: UMAP Visualization of Nearest Neighbors Retrieval.** The datastore consists of the training set (+WT) of all three datasets used in this work. Each colored dot represents a non-0 token from the training set. The embeddings are generated using JobBERTa. The orange shade represents the retrieved neighbors with  $k = 4$  for each token that is a skill (i.e., not an 0 token). Note that for the middle plot, the orange shade covers the blue clusters SAYFULLINA. GREEN has the green shade and SKILLSPAN are the darkcyan colors.

### 9.5.3 Prediction of Unseen Skills

The UMAP plots in [Figure 9.3](#) suggest that some datasets are closer to each other than others. To quantify this, we investigate the overlap of annotated skills between datasets and assess cross-dataset performance of NNOSE on unseen skills.

**Overlap of Datasets.** We calculate the exact span overlap of skills between the training sets of the datasets using the Jaccard similarity coefficient ([Jaccard, 1901](#)):  $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$ , where  $A$  and  $B$  are sets of multi-token spans (e.g., “manage a team”) from two separate training sets. The Jaccard similarity coefficients are as follows:  $J(\text{SKILLSPAN}, \text{SAYFULLINA}) = 0.35$ ,  $J(\text{SAYFULLINA}, \text{GREEN}) = 0.10$ , and  $J(\text{SKILLSPAN}, \text{GREEN}) = 0.29$ . These Jaccard coefficients indicate overlap between unique skill spans across datasets, suggesting that NNOSE can introduce the model to new and unseen skills.

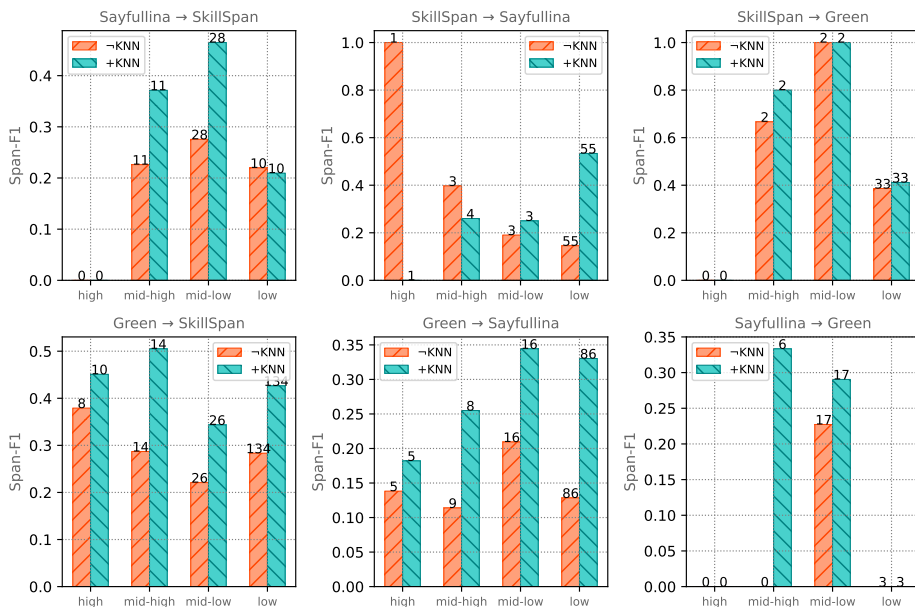


Figure 9.4: **Cross-dataset Long-tail Performance.** Similar to Figure 9.2, we plot the cross-dataset long-tail performance. NNOSE uses the datastore with all datasets. Training and evaluation data (test) are indicated in graph titles. Frequency bins are based on the training data span frequency; there are three frequency ranges: *high*: 10–15, *mid-high*: 7–10, *mid-low*: 4–6, *low*: 0–3.

**Results.** Table 9.3 presents the performance of JobBERTa across datasets. For completeness, we include a baseline where JobBERTa is fine-tuned on a union of all datasets (ALL). We notice training on the union of the data never leads to the best target dataset performance. Generally, we observe that in-domain data is best, both in vanilla and NNOSE setups (diagonal in Table 9.3). Performance drops when a model is applied to a dataset other than the one it was trained on (off-diagonal). Using NNOSE leads to substantial improvements across the challenging off-diagonal (cross-dataset) settings, while performance remains stable within datasets. We observe the largest improvements when applied to SAYFULLINA, with up to a 30% increase in span-F1. This is likely due to SAYFULLINA consisting

		↓Trained on	SKILLSPAN	SAYFULLINA	GREEN
Vanilla	SKILLSPAN			18.05	43.17
	SAYFULLINA	9.44			11.79
	GREEN	29.67	15.93		
	ALL	59.33	90.16	44.59	
+ $k$ NN	SKILLSPAN			45.86 ↑27.81	45.44 ↑2.27
	SAYFULLINA	26.16 ↑16.72			25.38 ↑13.59
	GREEN	41.22 ↑11.55	46.58 ↑30.65		
	ALL	59.51 ↑0.31	90.33 ↑0.17	45.63 ↑1.04	

Table 9.3: **Results of Unseen Skills based on JobBERTa ( $\forall D+WT$ ).** In the vanilla setting, models trained on one skill dataset are applied to another on test, showing varied performance. However, applying  $k$ NN improves the detection of unseen skills. Diagonal results can be found in Table 9.2. Refer to Table 9.10 for tuned hyperparameters.

mostly of soft skills, which are less prevalent in SKILLSPAN and GREEN, making it beneficial to introduce soft skills. Conversely, when the model is trained on SAYFULLINA, the absolute improvement on SKILLSPAN is lower, indicating that skill datasets can benefit each other to different extents.

#### 9.5.4 Cross-dataset Long-tail Analysis

Table 9.3 shows improvements when NNOSE is used in favor of vanilla fine-tuning. Figure 9.4 presents the long-tail performance analysis in the cross-dataset scenario, similar to Figure 9.2. We observe the largest gains with NNOSE in the low or mid-low frequency bins. However, exceptions are SKILLSPAN→GREEN and SAYFULLINA→GREEN, where most gains occur in the mid-high bin. Notably, SAYFULLINA→GREEN demonstrates higher performance with NNOSE, where all 6 skills are incorrectly predicted in the mid-high bin. An analysis of precision and recall in Table 9.12 (Section 9.9) substantiates that the improvements are both precision and recall-based, with gains of up to 40 recall points and 35.4 precision points in GREEN→SAYFULLINA. There is also an improvement up to 35.5 recall

	False Positives	False Negatives
SKILLSPAN	cleaning decisive Apache Camel building consumer demand for sustainable products	GCP IBM MQ AWS budget responsible
SAYFULLINA	empathy leadership management communication ability to manage and prioritise multiple assignments and tasks	leadership
GREEN	SQL scripting languages Manage a team troubleshooting activities dealing with tenants	software engineering development DevOps Cisco network administration

Table 9.4: **FPs & FNs of NNOSE.** We show several examples of false positives and false negatives in each dataset. We only show the predictions of NNOSE that are *not* in the vanilla model predictions.

points and 34.1 precision points for SKILLSPAN→SAYFULLINA. This further solidifies that memorizing tokens (i.e., storing all skills in the datastore) helps recall as mentioned in [Khandelwal et al. \(2020\)](#), and more importantly, highlighting the benefits of NNOSE in cross-dataset scenarios for SE.

### 9.5.5 Qualitative Check on Prediction Errors

We perform a qualitative analysis on the false positives (fp) and false negatives (fn) of NNOSE predictions compared to vanilla fine-tuning for each dataset. This analysis tells us whether a prediction corresponds to an actual skill, even if it does not contribute positively to the span-F1 metric. We observe that NNOSE produces a significant number of false positives that are “similar” to genuine skills. In [Table 9.4](#), for each dataset, we picked five fps and fns that represent hard, soft, and personal skills well (if applicable). We show the fps and fns for JobBERTa with NNOSE, we only show predictions that are *not* in the vanilla model predictions. In SAYFULLINA, there is only one fn. We notice from the errors, and especially the fps, that these are definitely skills, indicating the benefit of NNOSE helping to predict new skills or missed annotations.

## 9.6 Related Work

### 9.6.1 Skill Extraction

The dynamic nature of labor markets has led to an increase in tasks related to JD, including skill extraction (Kivimäki et al., 2013; Zhao et al., 2015; Sayfullina et al., 2018; Smith et al., 2019; Tamburri et al., 2020; Shi et al., 2020; Chernova, 2020; Bholā et al., 2020; Gugnani and Misra, 2020; Fareri et al., 2021; Konstantinidis et al., 2010; Zhang et al., 2022a,c,d; Green et al., 2022; Gnehm et al., 2022b; Beauchemin et al., 2022; Decorte et al., 2022; Ao et al., 2023; Goyal et al., 2023; Zhang et al., 2023b). These works employ methods such as sequence labeling (Sayfullina et al., 2018; Smith et al., 2019; Chernova, 2020; Zhang et al., 2022a,d), multi-label classification (Bholā et al., 2020), and graph-based methods (Shi et al., 2020; Goyal et al., 2023). Recent methodologies include domain-specific models where LMs are continuously pre-trained on unlabeled JD (Zhang et al., 2022a; Gnehm et al., 2022b). However, none of these methodologies introduce a retrieval-augmented model like NNOSE.

### 9.6.2 General Retrieval-augmentation

In retrieval augmentation, LMs can utilize external modules to enhance their context-processing ability. Two approaches are commonly used: First, using a separately trained model to retrieve relevant documents from a collection. This approach is employed in open-domain question answering tasks (Petroni et al., 2021) and with specific models such as ORQA (Lee et al., 2019), REALM (Guu et al., 2020), RAG (Lewis et al., 2020b), FiD (Izacard and Grave, 2021), and ATLAS (Izacard et al., 2022).

Second, previous work on explicit memorization showed promising results with a cache (Grave et al., 2017), which serves as a type of datastore. The cache contains past hidden states of the model as keys and the next word as tokens in key–value pairs. Memorization of hidden states in a datastore, involves using the  $k$ NN algorithm as the retriever. The first work of the  $k$ NN algorithm as the retrieval component was by Khandelwal et al.

(2020), leading to several LM decoder-based works.

### 9.6.3 Decoder-based Nearest Neighbor Approaches

Decoder-based nearest neighbors approaches are primarily focused on language modeling (Khandelwal et al., 2020; He et al., 2021; Yogatama et al., 2021; Ton et al., 2022; Shi et al., 2022; Jin et al., 2022; Bhardwaj et al., 2022; Xu et al., 2023) and machine translation (Khandelwal et al., 2021; Zheng et al., 2021; Jiang et al., 2021, 2022a; Wang et al., 2022a; Martins et al., 2022a,b; Zhu et al., 2022; Du et al., 2023; Zhu et al., 2023c; Min et al., 2023b,a). These approaches often prioritize efficiency and storage space reduction, as the datastores for these tasks can contain billions of tokens.

### 9.6.4 Encoder-based Nearest Neighbor Approaches.

Encoder-based nearest neighbor approaches have been explored in tasks such as named entity recognition (Wang et al., 2022b) and emotion classification (Yin and Shang, 2022). Here, the datastores are limited to single datasets with the sentence (or token) gold label pairs. Instead, we show the potential of adding multiple datasets in the datastore.

## 9.7 Conclusion

We introduce NNOSE, an LM that incorporates and leverages a non-parametric datastore for nearest neighbor retrieval of skill tokens. To the best of our knowledge, we are the first to introduce the nearest neighbors retrieval component for the extraction of occupational skills. We evaluated NNOSE on three relevant skill datasets with a wide range of skills and show that NNOSE enhances the performance of all LMs used in this work *without* additionally tuning the LM parameters. Through the combination of train sets in the datastore, our analysis reveals that NNOSE effectively leverages all the datasets by retrieving tokens from each. Moreover, NNOSE not only performs well on rare skills but also enhances

the performance of more frequent patterns. Lastly, we observe that our baseline models exhibit poor performance when applied in a cross-dataset setting. However, with the introduction of NNOSE, the models improve across all settings. Overall, our findings indicate that NNOSE is a promising approach for application-specific skill extraction setups and potentially helps discover skills that were missed in manual annotations.

## Limitations

We consider several limitations: One is the limited diversity of the datasets used in this work. Our study was constrained by the use of only three English datasets. By focusing solely on English data, we might have overlooked insights that exist in other languages. While these datasets were carefully selected to ensure relevance and quality, the limited scope of the data may restrict the generalizability of our findings to other SE datasets. Future research includes incorporating a wider range of datasets from diverse sources to obtain a more comprehensive understanding of the topic. Potential interesting future work should include validation on whether NNOSE works in a multilingual setting.

Another limitation is that we do skill detection and not specific labeling of the extracted spans, i.e., extracting generic B, I, O tags. This was to ensure that the datasets could be used in unison in the datastore. Interesting future work could extend NNOSE to include labeled skills in the datastore.

## Ethics Statement

The subject of job-related language models is a highly contentious topic, often sparking intense debates surrounding the issue of bias. We acknowledge that LMs such as JobBERTa and NNOSE possess the potential for inadvertent consequences, such as unconscious bias and dual-use when employed in the candidate selection process for specific job positions. There are research efforts to develop fairer recommender systems in the field of

human resources, focusing on mitigating biases (e.g., [Mujtaba and Mahapatra, 2019](#); [Raghavan et al., 2020](#); [Deshpande et al., 2020](#); [Köchling and Wehner, 2020](#); [Sánchez-Monedero et al., 2020](#); [Wilson et al., 2021](#); [van Els et al., 2022](#); [Arafan et al., 2022](#)). Nevertheless, one potential approach to alleviating such biases involves the retrieval of sparse skills for recall (e.g., this work). It is important to note, however, that we have not conducted an analysis to ascertain whether this particular method exacerbates any pre-existing forms of bias.



1	Experience	0
2	in	0
3	working	B
4	on	I
5	a	I
6	cloud-based	I
7	application	I
8	running	0
9	on	0
10	Docker	B
11	.	0
12	A	0
13	degree	B
14	in	I
15	Computer	I
16	Science	I
17	or	0
18	related	0
19	fields	0
20	fields	0
21	.	0

Figure 9.5: **Data Example for SkillSpan.** In SKILLSPAN, we would like to note the long skills.

1	ability	0
2	to	0
3	work	B
4	under	I
5	stress	I
6	condition	0
7		
8	due	0
9	to	0
10	the	0
11	dynamic	B
12	nature	0
13	of	0
14	the	0
15	group	0
16	environment	0
17	,	0
18	the	0
19	ideal	0
20	candidate	0
21	will	0

Figure 9.6: **Data Example for Sayfullina.** In SAYFULLINA, the skills are usually soft-like skills.

1	A	0
2	sound	0
3	understanding	0
4	of	0
5	the	0
6	Care	B
7	Standards	I
8	together	0
9	with	0
10	a	0
11	Nursing	B
12	qualification	I
13	and	0
14	current	0
15	NMC	B
16	registration	I
17	are	0
18	essential	0
19	for	0
20	this	0
21	role	0

Figure 9.7: **Data Example for Green.** There are many qualification skills (e.g., certificates).

## 9.8 Appendix

### 9.8.1 Whitening Transformation Algorithm

We apply the whitening transformation to the query embedding and the embeddings in the datastore. We can write a set of token embeddings as a set of row vectors:  $\{x_i\}_{i=1}^N$ . Additionally, a linear transformation  $\tilde{x}_i = (x_i - \mu)W$  is applied, where  $\mu = \frac{1}{N} \sum_{i=1}^N x_i$ . To obtain the matrix  $W$ , the following steps are conducted: First, we obtain the original covariance matrix

---

**Algorithm 4:** Whitening Transformation Workflow
 

---

```

1 input: Embeddings  $\{x_i\}_{i=1}^N$ ;
2 Compute  $\mu = \frac{1}{N} \sum_{i=1}^N x_i$  and  $\Sigma$  of  $\{x_i\}_{i=1}^N$ 
3 Compute  $U, \Lambda, U^\top = \text{SVD}(\Sigma)$ 
4 Compute  $W = U\sqrt{\Lambda^{-1}}$ 
5 for  $i = 1, 2, \dots, n$  do
6   |  $\tilde{x}_i = (x_i - \mu)W$ 
7 end
8 return  $\{\tilde{x}_i\}_{i=1}^N$ ;

```

---

$$\Sigma = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^\top (x_i - \mu). \quad (9.3)$$

Afterwards, we obtain the transformed covariance matrix  $\tilde{\Sigma} = W^\top \Sigma W$ , where we specify  $\tilde{\Sigma} = I$ . Therefore,  $\Sigma = (W^\top)^{-1} W^{-1} = (W^{-1})^\top W^{-1}$ . Here,  $\Sigma$  is a positive definite symmetric matrix that satisfies the following singular value decomposition (SVD; Golub and Reinsch, 1971) as indicated by Su et al. (2021):  $\Sigma = U\Lambda U^\top$ .  $U$  is an orthogonal matrix,  $\Lambda$  is a diagonal matrix, and the diagonal elements are all positive. Therefore, let  $W^{-1} = \sqrt{\Lambda}U^\top$ , we obtain the solution:  $W = U\sqrt{\Lambda^{-1}}$ . Putting it all together, as input, we have the set of embeddings  $\{x_i\}_{i=1}^N$ . We compute  $\mu$  and  $\Sigma$  of  $\{x_i\}_{i=1}^N$ . Then, we perform SVD on  $\Sigma$  to obtain matrices  $U$ ,  $\Lambda$ , and  $U^\top$ . Using these matrices, we calculate the transformation matrix  $W$ . Finally, we apply the transformation to each embedding in the set by subtracting  $\mu$  and multiplying by  $W$ . We are left with  $\tilde{x}_i = (x_i - \mu)W$ . Note that we do WT *before* we store the embedding in the datastore, and apply WT to the token embedding before we query the datastore.

We show the Whitening Transformation procedure in Algorithm 4. Note that Li et al. (2020b); Su et al. (2021) introduced a dimensionality reduction factor  $k$  on  $W$  ( $W[:, : k]$ ). The diagonal elements in the matrix  $\Lambda$  obtained from the SVD algorithm are in descending order. One can decide to keep the first  $k$  columns of  $W$  in line 6. This is similar to PCA

(Abdi and Williams, 2010). However, empirically, we found that reducing dimensionality had a negative effect on downstream performance, thus we omit that in this implementation.

## 9.8.2 Data Examples

SKILLSPAN	Figure 9.5
SAYFULLINA	Figure 9.6
GREEN	Figure 9.7

Table 9.5: Data example references for each dataset.

In Table 9.5, we refer to several listings of examples of the datasets. Notably in SKILLSPAN, the original samples contain two columns of labels. These refer to skills and knowledge. To accommodate for the approach of NNOSE, we merge the labels together and thus removing the possible nesting of skills. Zhang et al. (2022a) mentions that there is not a lot of nesting of skills. Following Zhang et al. (2022a), we prioritize the skills column when merging the labels. When there is nesting, we keep the labels of skills and remove the knowledge labels.

## 9.8.3 Implementation Details

### 9.8.4 General Implementation

We obtain all LMs from the Transformers library (Wolf et al., 2020b) and implement JobBERTa using the same library. All learning rates for fine-tuning are  $5 \times 10^{-5}$  using the AdamW optimizer (Loshchilov and Hutter, 2019). We use a batch size of 16 and a maximum sequence length of 128 with dynamic padding. The models are trained for 20 epochs with early stopping using a patience of 5. We implement the retrieval component using the FAISS library (Johnson et al., 2019), which is a standard for nearest neighbors retrieval-augmented methods.<sup>3</sup>

<sup>3</sup><https://faiss.ai/>

Dataset →		SKILLSPAN	SAYFULLINA	GREEN
JobBERT	$k$	4	4	16
	$\lambda$	0.3	0.3	0.15
	$T$	0.1	2.0	10.0
RoBERTa	$k$	32	4	64
	$\lambda$	0.3	0.3	0.25
	$T$	10.0	0.1	10.0
JobBERTa	$k$	16	4	8
	$\lambda$	0.2	0.1	0.1
	$T$	5.0	10.0	10.0
Search Space	$k$	{4, 8, 16, 32, 64, 128}		
	$\lambda$	{0.1, 0.15, 0.2, 0.25, ..., 0.9}		
	$T$	{0.1, 0.5, 1.0, 2.0, 3.0, 5.0, 10.0}		

Table 9.6: Tuned Hyperparameters on Dev. These are for {D}.

Dataset →		SKILLSPAN	SAYFULLINA	GREEN
JobBERT	$k$	4	16	32
	$\lambda$	0.3	0.25	0.15
	$T$	10.0	5.0	10.0
RoBERTa	$k$	16	8	8
	$\lambda$	0.15	0.1	0.1
	$T$	10.0	10.0	10.0
JobBERTa	$k$	8	4	8
	$\lambda$	0.2	0.15	0.1
	$T$	0.5	0.1	10.0
Search Space	$k$	{4, 8, 16, 32, 64, 128}		
	$\lambda$	{0.1, 0.15, 0.2, 0.25, ..., 0.9}		
	$T$	{0.1, 0.5, 1.0, 2.0, 3.0, 5.0, 10.0}		

Table 9.8: Tuned Hyperparameters on Dev. These are for  $\forall D$ .

### 9.8.5 JobBERTa

We apply domain-adaptive pre-training (Gururangan et al., 2020b), which involves continued self-supervised pre-training of a large LM on domain-specific text. This approach enhances the modeling of text for downstream tasks within the domain. We continue pre-training on a roberta-base checkpoint with 3.2M job posting sentences from Zhang et al. (2022a). We use a batch size of 8 and run MLM for a single epoch following Gururangan et al. (2020b). The rest of the hyperparameters are set to the defaults in the Transformer library.<sup>4</sup>

<sup>4</sup>[https://github.com/huggingface/transformers/blob/main/examples/pytorch/language-modeling/run\\_mlm.py](https://github.com/huggingface/transformers/blob/main/examples/pytorch/language-modeling/run_mlm.py)

Dataset →		SKILLSPAN	SAYFULLINA	GREEN
JobBERT	$k$	4	4	64
	$\lambda$	0.35	0.35	0.4
	$T$	2.0	0.1	5.0
RoBERTa	$k$	32	4	16
	$\lambda$	0.35	0.45	0.25
	$T$	0.1	0.1	1.0
JobBERTa	$k$	64	128	128
	$\lambda$	0.25	0.35	0.45
	$T$	10.0	0.5	10.0
Search Space	$k$	{4, 8, 16, 32, 64, 128}		
	$\lambda$	{0.1, 0.15, 0.2, 0.25, ..., 0.9}		
	$T$	{0.1, 0.5, 1.0, 2.0, 3.0, 5.0, 10.0}		

Table 9.7: Tuned Hyperparameters on Dev. These are for {D}+WT.

Dataset →		SKILLSPAN	SAYFULLINA	GREEN
JobBERT	$k$	32	4	128
	$\lambda$	0.3	0.3	0.4
	$T$	1.0	0.5	2.0
RoBERTa	$k$	128	128	64
	$\lambda$	0.35	0.1	0.25
	$T$	0.1	0.5	0.1
JobBERTa	$k$	32	8	128
	$\lambda$	0.15	0.3	0.2
	$T$	0.1	0.1	2.0
Search Space	$k$	{4, 8, 16, 32, 64, 128}		
	$\lambda$	{0.1, 0.15, 0.2, 0.25, ..., 0.9}		
	$T$	{0.1, 0.5, 1.0, 2.0, 3.0, 5.0, 10.0}		

Table 9.9: Tuned Hyperparameters on Dev. These are for  $\forall D$ +WT.

↓Trained on	Hyperparams.	SKILLSPAN	SAYFULLINA	GREEN
SKILLSPAN	$k$		16	32
	$\lambda$		0.9	0.7
	$T$		0.1	0.5
SAYFULLINA	$k$	64		32
	$\lambda$	0.9		0.8
	$T$	0.1		0.1
GREEN	$k$	32	32	
	$\lambda$	0.85	0.9	
	$T$	0.5	0.1	
ALL	$k$	4	128	32
	$\lambda$	0.25	0.6	0.65
	$T$	1.0	1.0	0.5
Search Space	$k$	{4, 8, 16, 32, 64, 128}		
	$\lambda$	{0.1, 0.15, 0.2, 0.25, ..., 0.9}		
	$T$	{0.1, 0.5, 1.0, 2.0, 3.0, 5.0, 10.0}		

Table 9.10: **Results of Unseen Skills (Development Set) based on JobBERTa.**

### 9.8.6 NNOSE Setup

Following previous work, the keys used in NNOSE are the 768-dimensional representation logits obtained from the final layer of the LM (input to the softmax). We perform a single forward pass over the training set of each dataset to save the keys and values, i.e., the hidden representation and the corresponding gold BIO tag. The FAISS index is created using all the keys to learn 4096 cluster centroids. During inference, we retrieve  $k$  neighbors. The index looks up 32 cluster centroids while searching for the nearest neighbors. For all experiments, we compute the squared Euclidean ( $L^2$ ) distances with full precision keys. The difference in inference speed is almost negligible, with the  $k$ NN module taking a few extra seconds compared to regular inference. For the exact hyperparameter values, we indicate them in the next paragraph.

Dataset (Dev.) →	Setting	SKILLSPAN	SAYFULLINA	GREEN	avg. Span-F1
JobBERT (Zhang et al., 2022a)		61.08	89.26	37.27	62.54
+ $k$ NN	{D}	61.56 $\uparrow$ 0.48	89.69 $\uparrow$ 0.43	37.48 $\uparrow$ 0.21	62.91 $\uparrow$ 0.37
+ $k$ NN	{D}+WT	61.77 $\uparrow$ 0.69	89.78 $\uparrow$ 0.52	38.07 $\uparrow$ 0.80	63.21 $\uparrow$ 0.67
+ $k$ NN	$\forall$ D	61.58 $\uparrow$ 0.50	89.50 $\uparrow$ 0.24	37.27 $-$ 0.00	62.78 $\uparrow$ 0.24
+ $k$ NN	$\forall$ D+WT	61.50 $\uparrow$ 0.42	89.37 $\uparrow$ 0.11	38.19 $\uparrow$ 0.92	63.02 $\uparrow$ 0.48
RoBERTa (Liu et al., 2019)		65.02	92.91	40.33	66.09
+ $k$ NN	{D}	65.36 $\uparrow$ 0.34	92.76 $\downarrow$ 0.15	40.53 $\uparrow$ 0.20	66.22 $\uparrow$ 0.13
+ $k$ NN	{D}+WT	65.34 $\uparrow$ 0.32	93.07 $\uparrow$ 0.16	41.22 $\uparrow$ 0.89	66.54 $\uparrow$ 0.45
+ $k$ NN	$\forall$ D	64.98 $\downarrow$ 0.04	92.78 $\downarrow$ 0.13	40.60 $\uparrow$ 0.27	66.12 $\uparrow$ 0.03
+ $k$ NN	$\forall$ D+WT	65.38 $\uparrow$ 0.36	92.92 $\uparrow$ 0.01	41.11 $\uparrow$ 0.77	66.47 $\uparrow$ 0.38
JobBERTa (This work)		65.15	92.09	40.59	65.94
+ $k$ NN	{D}	65.25 $\uparrow$ 0.10	91.99 $\downarrow$ 0.10	41.31 $\uparrow$ 0.72	66.18 $\uparrow$ 0.24
+ $k$ NN	{D}+WT	65.21 $\uparrow$ 0.06	92.10 $\uparrow$ 0.01	41.41 $\uparrow$ 0.82	66.24 $\uparrow$ 0.30
+ $k$ NN	$\forall$ D	65.15 $-$ 0.00	92.04 $\downarrow$ 0.05	40.83 $\uparrow$ 0.24	66.01 $\uparrow$ 0.07
+ $k$ NN	$\forall$ D+WT	65.22 $\uparrow$ 0.07	92.13 $\uparrow$ 0.04	41.45 $\uparrow$ 0.86	66.26 $\uparrow$ 0.32

**Table 9.11: Development Set Results.** There are four settings for each model. {D}: in-dataset datastore (i.e., the datastore only contains the keys from the specific training data it is applied on).  $\forall$ D: The datastore contains the keys from all available training datasets. + $W$ : Whitening Transformation is applied to the keys before adding them to the datastore or querying the datastore. We indicate the performance increase ( $\uparrow$ ), decrease ( $\downarrow$ ), or no change ( $-$ ) when using  $k$ NN compared to not using  $k$ NN. Additionally, we show the average span-F1 performance of each model across the three datasets. In the development set, it seems that an in-dataset datastore works best.

### 9.8.7 Hyperparameters NNOSE

The best-performing hyperparameters and search space can be found in Table 9.6, Table 9.7, Table 9.8, and Table 9.9. We report the  $k$ -nearest neighbors,  $\lambda$  value, and softmax temperature  $T$  for each dataset and model.

In Table 9.10, we show the hyperparameters for the cross-dataset analysis. In the vanilla setting, we apply the models trained on a particular skill dataset to another skill dataset, similar to transfer learning. We observe a significant discrepancy in performances cross-dataset, indicating a wide range of skills. However, when  $k$ NN is applied, it improves the detection of unseen skills. The datastore contains tokens from all datasets.

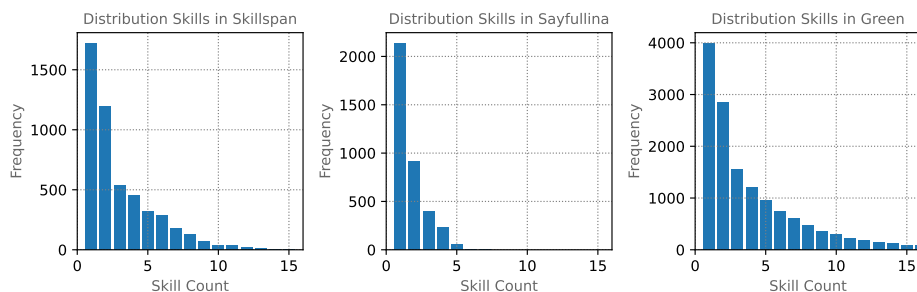


Figure 9.8: **Frequency Distribution of Skill Occurrences in the Train Set.** We display the frequency distribution of skill occurrences in each train set. *How to read:* For instance, in the case of Sayfullina, there are over 2,000 skills that occur only **once** in the training set. We demonstrate that all skill datasets exhibit an inherent long-tail pattern.

### 9.8.8 Development Set Results

We show the dev. set results in [Table 9.11](#). Overall, the patterns of improvements hold across datasets and models. We base the test set result on the best-performing setups in the development set, i.e.,  $\{D\}+WT$  and  $\forall D+WT$ .

### 9.8.9 Frequency Distribution of Skills

We show the skill frequency distribution of the datasets in [Figure 9.8](#), as mentioned in [Subsection 9.5.1](#). Here, we show evidence of the long-tail pattern in skills for each dataset. There is a cut-off at count 15 for GREEN, indicating that there are skills in the development set that occur more than 15 times.

### 9.8.10 Qualitative Results NNOSE

We show several qualitative results of NNOSE. In [Table 9.13](#), we show a qualitative sample of using JobBERTa on SKILLSPAN. The current token is “IT” with gold label 0. The language model puts 0.4 softmax probability on the tag I. By retrieving the nearest neighbors, the final probability mass gets shifted towards 0 with probability 0.43, which is the correct tag.

Setup↓	Vanilla		+ $k$ NN	
	Precision	Recall	Precision	Recall
SAYFULLINA→SKILLSPAN	10.20	10.50	37.67↑27.47	29.62↑19.12
GREEN→SKILLSPAN	28.40	33.56	46.00↑11.60	46.29↑12.73
SKILLSPAN→SAYFULLINA	15.19	23.42	49.25↑34.06	58.95↑35.53
GREEN→SAYFULLINA	12.80	21.58	48.21↑35.41	61.87↑40.29
SKILLSPAN→GREEN	52.01	37.42	55.37↑3.36	38.74↑1.32
SAYFULLINA→GREEN	17.79	7.64	39.83↑22.04	18.31↑10.67

Table 9.12: **Precision & Recall Numbers Cross-dataset on Test.** We show the precision and recall numbers in the cross-dataset setup. We use the  $\forall D+WT$  setup here, with JobBERTa as the backbone model.

In Table 9.14, we show a qualitative sample of using JobBERTa on SKILLSPAN with multi-token annotations and how this behaves. The current skill is “coding skills” with gold labels B and I respectively. Both the model and  $k$ NN puts high confidence in the correct label. Note that the nearest neighbors of “coding” are quite varied, which shows the benefit of NNOSE. Note that all the retrieved “skills” tokens are from different contexts.

In Table 9.15, we show a qualitative sample of using JobBERTa on SKILLSPAN. The current token is “optimistic” with gold label B. This is a so-called “soft skill”. The language model puts high confidence in the tag B, which is the correct tag. The retrieved neighbors are frequently relevant, but sometimes less. This indicates that the retrieved neighbors (all soft skills) occur in similar contexts.

In Table 9.16, we show a qualitative sample of using JobBERTa on SKILLSPAN. The current token is “optimistic” with gold label B. This is a so-called “soft skill”. The language model puts high confidence in the tag B, which is the correct tag. The retrieved neighbors are frequently relevant, but sometimes less. This indicates that the retrieved neighbors (all soft skills) occur in similar contexts.



JobBERTa $\rightarrow$ SKILLSPAN	
Current token	IT
Gold label	0
LM prediction probs	[0.277, 0.404, 0.319]
Nearest neighbors ( $k = 8$ )	['IT', 'Software', 'Software', 'Cloud', 'Cloud', 'Database', 'Ag', 'software']
Aggregated $k$ NN scores	[0.000, 0.132, 0.868]
Final predicted probs	[0.221, 0.350, 0.429]

Table 9.13: **Cherry Picked Qualitative Sample NNOSE of Higher Precision.** We show a qualitative sample of using JobBERTa on SKILLSPAN. In this case, we see more weight being put on a specific tag, resulting in higher precision.

## 9.9 Further Cross-dataset Analysis

In Table 9.12, we checked the precision and recall numbers for the cross-dataset setup with  $\forall D+WT$  and JobBERTa as the backbone model. When using NNOSE, we generally notice an increase in precision, with the largest when applied to SAYFULLINA. The largest gains are with respect to recall, we notice a significant gain in all setups, where the recall and precision increase is mixed. This indicates that NNOSE is a useful method for both precision-focused and recall-focused applications, as we are storing skills in the datastore to be retrieved.

JobBERTa → SKILLSPAN	
Current token	coding
Gold label	B
LM prediction probs	[0.988, 0.000, 0.012]
Nearest neighbors ( $k = 8$ )	['programming', 'coding', 'programming', 'debugging', 'scripting', 'writing', 'coding', 'programming']
Aggregated $k$ NN scores	[1.000, 0.000, 0.000]
Final predicted probs	[0.991, 0.000, 0.009]
Current token	skills
Gold label	I
LM prediction probs	[0.000, 0.990, 0.010]
Nearest neighbors ( $k = 8$ )	['skills', 'skills', 'skills', 'skills', 'skills', 'skills', 'skills', 'skills']
Aggregated $k$ NN scores	[0.000, 1.000, 0.000]
Final predicted probs	[0.000, 0.992, 0.008]

Table 9.14: **Cherry Picked Qualitative Sample NNOSE of Multiple Tokens.** We show a qualitative sample of using JobBERTa on SKILLSPAN with multi-token annotations and how this behaves.

JobBERTa → GREEN	
Current token	tools
Gold label	I
LM prediction probs	[0.250, 0.374, 0.379]
Nearest neighbors ( $k = 8$ )	['tools', 'tools', 'transport', 'transport', 'transport', 'transport', 'car', 'transport']
Aggregated $k$ NN scores	[0.124, 0.626, 0.250]
Final predicted probs	[0.234, 0.399, 0.366]

Table 9.15: **Cherry Picked Qualitative Sample NNOSE of Randomness.** We show a qualitative sample of using JobBERTa on SKILLSPAN. The language model puts high confidence on the tag I, which is the correct tag. Here the retrieved neighbors do not seem too relevant, which in this case is mostly random chance that it got it correctly.

JobBERTa → SKILLSPAN	
Current token	optimistic
Gold label	B
LM prediction probs	[0.998, 0.000, 0.002]
Nearest neighbors ( $k = 8$ )	['proactive', 'responsible', 'holistic', 'operational', 'positive', 'open', 'professional', 'agile']
Aggregated $k$ NN scores	[1.000, 0.000, 0.000]
Final predicted probs	[0.999, 0.000, 0.001]

Table 9.16: **Cherry Picked Qualitative Sample NNOSE of Variety.** We show a qualitative sample of using JobBERTa on SKILLSPAN. The language model puts high confidence in the tag B, which is the correct tag. The retrieved neighbors are frequently relevant.



## **Part IV**

# **Linking Skills to Existing Sources**



# Chapter 10

## Entity Linking in the Job Market Domain

The work presented in this chapter is based on a paper that has been accepted as: Mike Zhang, Rob van der Goot, and Barbara Plank. Entity linking in the job market domain. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 410–419, St. Julian’s, Malta, March 2024b. Association for Computational Linguistics. URL <https://aclanthology.org/2024.findings-eacl.28>.

## Abstract

In Natural Language Processing, entity linking (EL) has centered around Wikipedia, but yet remains underexplored for the job market domain. Disambiguating skill mentions can help us get insight into the current labor market demands. In this work, we are the first to explore EL in this domain, specifically targeting the linkage of occupational skills to the ESCO taxonomy (le Vrang et al., 2014). Previous efforts linked coarse-grained (full) sentences to a corresponding ESCO skill. In this work, we link more fine-grained span-level mentions of skills. We tune two high-performing neural EL models, a bi-encoder (Wu et al., 2020) and an autoregressive model (Cao et al., 2021), on a synthetically generated mention–skill pair dataset and evaluate them on a human-annotated skill-linking benchmark. Our findings reveal that both models are capable of linking implicit mentions of skills to their correct taxonomy counterparts. Empirically, BLINK outperforms GENRE in strict evaluation, but GENRE performs better in loose evaluation (accuracy@k).<sup>1</sup>

## 10.1 Introduction

Labor market dynamics, influenced by technological changes, migration, and digitization, have led to the availability of job descriptions (JD) on platforms to attract qualified candidates (Brynjolfsson and McAfee, 2011, 2014; Balog et al., 2012). It is important to extract and link surface form skills to a unique taxonomy entry, allowing us to quantify the current labor market dynamics and determine the demands and needs. We attempt to tackle the problem of *entity linking* (EL) in the job market domain, specifically the linking of fine-grained span-level skill mentions to a specific taxonomy entry.

Generally, EL is the task of linking mentions of entities in unstructured text documents to their respective unique entities in a knowledge base

---

<sup>1</sup>The source code can be found at [https://github.com/jjzha/el\\_esco](https://github.com/jjzha/el_esco).



(KB), most commonly Wikipedia (He et al., 2013). Recent models address this problem by producing entity representations from a (sub)set of KB information, e.g., entity descriptions (Logeswaran et al., 2019; Wu et al., 2020), fine-grained entity types (Raiman and Raiman, 2018; Onoe and Durrett, 2020; Ayoola et al., 2022), or generation of the input text autoregressively (Cao et al., 2021, 2022).

For skill linking specifically, we use the European Skills, Competences, Qualifications and Occupations (ESCO; le Vrang et al., 2014) taxonomy due to its comprehensiveness. Previous work classified spans to its taxonomy code via multi-class classification (Zhang et al., 2022b) without surrounding context and neither the full breadth of ESCO. Gnehm et al. (2022a) approaches it as a sequence labeling task, but only uses more coarse-grained ESCO concepts, and not the full taxonomy. Last, others attempt to match the full sentence to their respective taxonomy title (Decorte et al., 2022, 2023; Clavié and Soulié, 2023).

The latter comes with a limitation: The taxonomy title does not indicate which subspan in the sentence it points to, without an exact match. We define this as an *implicit* skill, where mentions (spans) in the sentence do not have an exact string match with a skill in the ESCO taxonomy. The differences can range from single tokens to entire phrases. For example, we can link “being able to work together” to “plan teamwork”.<sup>2</sup> If we know the exact span, this implicit skill can be added to the taxonomy as an alternative choice for the surface skill. As a result, this gives us a more nuanced view of the labor market skill demands. Therefore, we attempt to train models to the linking of both implicit and explicit skill mentions.

**Contributions.** Our findings can be summarized as follows: ① We pose the task of skill linking as an entity linking problem, showing promising results of successful linking with two entity linking systems. ② We present a qualitative analysis showing that the model successfully links implicit skills to their respective skill entry in ESCO.

---

<sup>2</sup><https://t.ly/3VUJG>

	Instances	Unique Titles	UNK
Train	123,619	12,984	14,641
Dev.	480	149	233
Test	1,824	455	813

Table 10.1: **Data Statistics.** Data distribution of train, dev, and test splits. UNK indicates skills mentions that are not linked to a corresponding taxonomy title.

## 10.2 Methodology

### 10.2.1 Definition

In EL, we process the input document  $\mathcal{D} = \{w_1, \dots, w_r\}$ , a collection of entity mentions denoted as  $\mathcal{MD} = \{m_1, \dots, m_n\}$ , and a KB, ESCO in our case:  $\mathcal{E} = \{e_1, \dots, e_{13890}, \text{UNK}\}$ . The objective of an EL model is to generate a list of mention-entity pairs  $\{(m_i, e_i)\}_{i=1}^n$ , where each entity  $e$  corresponds to an entry in a KB. We assume that both the titles and descriptions of the entities are available, which is a common scenario in EL research (Ganea and Hofmann, 2017; Logeswaran et al., 2019; Wu et al., 2020). We also assume that each mention in the document has a corresponding valid gold entity present in the knowledge base, including UNK. This scenario is typically referred to as “in-KB evaluation”. Similar to prior research efforts (Logeswaran et al., 2019; Wu et al., 2020), we also presuppose that the mentions within the document have already been tagged.

### 10.2.2 Data

We use ESCO titles as ground truth labels, containing 13,890 skills.<sup>3</sup> Table 10.1 presents the train, dev, and test data in our experiments. We leverage the train set introduced by Decorte et al. (2023)<sup>4</sup> along with the dev and test sets provided in Decorte et al. (2022).<sup>5</sup> The train set is synthet-

<sup>3</sup>Per version 1.1.1, accessed on 01 August 2023.

<sup>4</sup><https://t.ly/edqkp>

<sup>5</sup><https://t.ly/LcqQ7>

ically generated by [Decorte et al. \(2023\)](#) with the `gpt-3.5-turbo-0301` model ([OpenAI, 2023](#)). Specifically, this involves taking each skill from ESCO and prompting the model to generate sentences resembling JD sentences that require that particular skill. The dev and test splits, conversely, are derived from actual job advertisements sourced from the study by [Zhang et al. \(2022a\)](#). These JDs are annotated with spans corresponding to specific skills, and these spans have subsequently been manually linked to ESCO, as described in the work of [Decorte et al. \(2022\)](#). In cases where skills cannot be linked, two labels are used, namely `UNDERSPECIFIED` and `LABEL NOT PRESENT`. For the sake of uniformity, we map both of these labels to a generic `UNK` tag. We used several heuristics based on Levenshtein distance and sentence similarity to find the exact subspans if it exceeds certain thresholds, otherwise, it is `UNK`. This process is outlined in [Subsection 10.6.1](#). In addition, some data examples can be found in [Subsection 10.6.2](#). The number of UNKs in the data is also in [Table 10.1](#). During inference, the `UNK` title is a prediction option for the models.

### 10.2.3 Models

We use two EL models, selected for their robust performance in EL on Wikipedia.<sup>6</sup>

**BLINK ([Wu et al., 2020](#)).** BLINK uses a bi-encoder architecture based on BERT ([Devlin et al., 2019a](#)), for modeling pairs of mentions and entities. The model processes two inputs:

$$[\text{CLS}] \text{ctxt}_l [\text{S}] \text{mention} [\text{E}] \text{ctxt}_r [\text{SEP}]$$

Where “mention”, “`ctxtl`”, and “`ctxtr`” corresponds to the wordpiece tokens of the mention, the left context, and the right context. The mention is denoted by special tokens `[S]` and `[E]`. The entity and its description are

---

<sup>6</sup>For the hyperparameter setups, we refer to [Subsection 10.6.3](#).

structured as follows:

[CLS] title [ENT] description [SEP]

Here, “title” and “description” represent the wordpiece tokens of the skills’s title and description, respectively. [ENT] is a special token to separate the two representations. We train the model to maximize the dot product of the [CLS] representation of the two inputs, for the correct skill in comparison to skills within the same batch. For each training pair  $(m_i, e_i)$ , the loss is computed as

$$\mathcal{L}(m_i, e_i) = -s(m_i, e_i) + \log \sum_{j=1}^B \exp(s(m_i, e_j)), \quad (10.1)$$

where the objective is to minimize the distance between  $m_i$  and  $e_i$  while encouraging the model to assign a higher score to the correct pair and lower scores to randomly sampled incorrect pairs. Hard negatives are also used during training, these are obtained by finding the top 10 predicted skills for each training example. These extra hard negatives are added to the random in-batch negatives.

**GENRE (Cao et al., 2021).** GENRE formulates EL as a retrieval problem using a sequence-to-sequence model based on BART (Lewis et al., 2020a). This model generates textual entity identifiers (i.e., skill titles) and ranks each entity  $e \in \mathcal{E}$  using an autoregressive approach:

$$s(e | x) = p_\theta(y | x) = \prod_{i=1}^N p_\theta(y_i | y_{<i}, x), \quad (10.2)$$

where  $y$  represents the set of  $N$  tokens in the identifier of entity  $e$  (i.e., entity tile), and  $\theta$  denotes the model parameters. During decoding, the model uses a constrained beam search to ensure the generation of valid identifiers (i.e., only producing valid titles that exist within the KB, including UNK).

	Train Source	Acc@1	Acc@4	Acc@8	Acc@16	Acc@32
Random		0.22 $\pm$ 0.00	0.88 $\pm$ 0.00	1.76 $\pm$ 0.00	3.52 $\pm$ 0.00	7.04 $\pm$ 0.00
TF-IDF		2.25 $\pm$ 0.00				
BLINK (bert-base)	ESCO	12.74 $\pm$ 0.49	22.81 $\pm$ 0.79	27.70 $\pm$ 0.82	32.44 $\pm$ 1.33	36.46 $\pm$ 1.07
BLINK (bert-large)	ESCO	12.77 $\pm$ 0.94	22.58 $\pm$ 1.47	27.24 $\pm$ 1.23	31.75 $\pm$ 0.89	36.10 $\pm$ 1.28
BLINK (bert-large)	Wiki (0-shot)	23.30 $\pm$ 0.00	<b>32.89<math>\pm</math>0.00</b>	<b>38.16<math>\pm</math>0.00</b>	42.60 $\pm$ 0.00	45.56 $\pm$ 0.00
BLINK (bert-large)	Wiki + ESCO	<b>23.55<math>\pm</math>0.14</b>	32.63 $\pm$ 0.16	37.38 $\pm$ 0.09	<b>43.25<math>\pm</math>0.13</b>	<b>48.98<math>\pm</math>0.21</b>
GENRE (bart-base)	ESCO	1.47 $\pm$ 0.05	4.84 $\pm$ 1.74	10.46 $\pm$ 6.81	11.30 $\pm$ 4.18	15.51 $\pm$ 4.62
GENRE (bart-large)	ESCO	2.33 $\pm$ 0.44	5.74 $\pm$ 1.43	8.18 $\pm$ 2.21	11.13 $\pm$ 2.42	15.26 $\pm$ 2.66
GENRE (bart-large)	Wiki (0-shot)	6.91 $\pm$ 0.00	12.34 $\pm$ 0.00	15.52 $\pm$ 0.00	21.60 $\pm$ 0.00	33.17 $\pm$ 0.00
GENRE (bart-large)	Wiki + ESCO	<b>11.48<math>\pm</math>0.41</b>	<b>21.26<math>\pm</math>0.43</b>	<b>27.40<math>\pm</math>0.78</b>	<b>37.21<math>\pm</math>0.69</b>	<b>49.78<math>\pm</math>1.05</b>

Table 10.2: **Skill Linking Results.** We show the results of the various models used. There are two base and four large models. Training sources are either ESCO or a combination of Wikipedia and ESCO. The results are the average and standard deviation over five seeds. For the 0-shot setup, we apply the fine-tuned models from the work of [Wu et al. \(2020\)](#) and [Cao et al. \(2021\)](#) to the ESCO test set once. We have a random and TF-IDF-based baseline.

## 10.2.4 Setup

We train a total of six models: for BLINK, these are BERT<sub>base</sub> and BERT<sub>large</sub> (uncased; [Devlin et al., 2019a](#)) trained on ESCO, and another large version trained on Wikipedia and ESCO sequentially. GENRE has the same setup, but then with BART ([Lewis et al., 2020a](#)). Additionally, we apply the released models from both BLINK and GENRE (large, trained on Wikipedia) in a zero-shot manner and evaluate their performance. The reason we use Wikipedia-based models is that we hypothesize this is due to many skills in ESCO also having corresponding Wikipedia pages (e.g., Python<sup>7</sup> or teamwork<sup>8</sup>), thus could potentially help linking. Next, to address unknown entities (UNK), we include them as possible label outputs. Last, our evaluation metric is Accuracy@*k*, following prior research ([Logeswaran et al., 2019](#); [Wu et al., 2020](#); [Zaporojets et al., 2022](#)).

<sup>7</sup>[https://en.wikipedia.org/wiki/Python\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))

<sup>8</sup><https://en.wikipedia.org/wiki/Teamwork>

## 10.3 Results

[Table 10.2](#) presents the results. Each model is trained for five seeds, and we report the average with standard deviation. We make use of a random and TF-IDF-based baseline. Firstly, we observe that the strict linking performance (i.e.,  $\text{Acc}@1$ ) is rather modest for both BLINK and GENRE. But most models outperform the baselines. Notably, the top-performing models in this context are the  $\text{BERT}_{\text{large}}$  and  $\text{BART}_{\text{large}}$  models, which were further fine-tuned from Wikipedia EL with ESCO. As expected, scores improve considerably as we increase the value of  $k$ . Secondly, for both BLINK and GENRE, model size seems not to have a substantial impact when trained only on ESCO. Specifically for BLINK, the performance remains consistent for  $\text{Acc}@1$  and exhibits only a slight decline as we relax the number of candidates for performance evaluation. For GENRE, the observed trend remains largely unchanged, even with a larger  $k$ .

Remarkably, the zero-shot setup performance of both BLINK and GENRE, when trained on Wikipedia, surpasses that of models trained solely on ESCO. For Wikipedia-based evaluation, GENRE usually outperforms BLINK. We notice the opposite in this case. For BLINK, this improvement is approximately 11 accuracy points for  $k = 1$ . Meanwhile, for GENRE, we observe an increase of roughly 9 accuracy points when trained on both Wikipedia and ESCO. This trend persists for a larger  $k$ , reaching up to a 12.5 accuracy point improvement for BLINK and a 34 accuracy point improvement for GENRE in the case of  $\text{Acc}@32$ . Furthermore, we show that further fine-tuning the Wikipedia-trained models on ESCO contributes to an improved EL performance at  $k = \{1, 16, 32\}$  for both models. For UNK-specific results, we refer to [Subsection 10.6.4](#). We confirm our hypothesis that Wikipedia has concepts that are also in ESCO, this gives the model strong prior knowledge of skills.

Mention	BLINK	GENRE
① Work in a way that is <b>patient-centred</b> and inclusive.	person centred care (K0913)	work in an organised manner (T)
② You can <b>ride a bike</b> .	sell bicycles (S1.6.1)	drive two-wheeled vehicles (S8.2.2)
③ It is expected that you are a super user of the <b>MS office tools</b> .	use Microsoft Office (S5.6.1)	tools for software configuration management (0613)
④ <b>Picking and packing</b> .	carry out specialised packing for customers (S6.1.3)	perform loading and unloading operations (S6.2.1)
⑤ You are expected to be able to further <b>develop your team</b> - both personally and professionally. <b>GOLD: manage a team (S4.8.1)</b>	manage personal professional development (S1.14.1)	shape organisational teams based on competencies (S4.6.0)
⑥ Our games are developed using Unity so we expect all our programmers to have solid knowledge of mobile game development in Unity3D and <b>C#</b> .	C# (K0613)	C# (K0613)

Table 10.3: We show six qualitative examples. The mention is indicated with purple and we show the predictions ( $k = 1$ ) of BLINK and GENRE. Green predictions mean correct, and red indicates wrong linking with respect to the ground truth. We also show the ESCO ID, indicating the differences in concepts. The results show successful linking of implicit mentions of skills. In example (5), we show how the linked results are still valid while being different concepts. However, evaluation does not count it as a correct hit.

## 10.4 Discussion

### 10.4.1 Qualitative Analysis

We manually inspected a subset of the predictions. We present qualitative examples in [Table 10.3](#). We found the following trends upon inspection:

- The EL models exhibit success in linking implicit and explicit men-

tions to their respective taxonomy titles (e.g., ①, ②, ④, ⑥).

- In cases of hard skills (③, ⑥), BLINK correctly matches “MS office tools” to “using Microsoft Office”, which is not an exact match. Both models predict the explicit mention “C#” correctly to the C# taxonomy title.
- We found that the models predict paraphrased versions of skills that could also be considered correct (④, ⑤), even being entirely different concepts (i.e., different ESCO IDs).

### 10.4.2 Evaluation Limitation

We qualitatively demonstrate the linking of skills that are implicit and/or valid. Empirically, we observe that the strict linking of skills leads to an underestimation of model performance. We believe this limitation is rooted in evaluation. In train, dev, and test, there is only *one* correct gold label. We reciprocate the findings by Li et al. (2020a), where they found that a large number of predictions are “technically correct” but limitations in Wikipedia-based evaluation falsely penalized their model (i.e., a more or less precise version of the same entity). Especially ⑤ in Table 10.3 shows this challenge for ESCO, we can consider multiple links to be correct for a mention given a particular context. This highlights the need for appropriate EL evaluation sets, not only for ESCO, but for EL in general.

## 10.5 Conclusion

We present entity linking in the job market domain, using two existing high-performing neural models. We demonstrate that the bi-encoder architecture of BLINK is more suited to the job market domain compared to the autoregressive GENRE model. While strict linking results favor BLINK over GENRE, if we relax the number of candidates, we observe that GENRE performs slightly better. From a qualitative perspective, the performance of strict linking results is modest due to limitations in the evaluation set,



which considers only one skill correct per mention. However, upon examining the predictions, we identify valid links, suggesting the possibility of multiple correct links for a particular mention, highlighting the need for more comprehensive evaluation. We hope this work sparks interest in entity linking within the job market domain.

## Limitations

In the context of EL for ESCO, our approach has several limitations. Firstly, it only supports English, and might not generalize to other languages. Secondly, our EL model is trained on synthetic training data, which may not fully capture the intricacies and variations present in real-world documents. The use of synthetic data could limit its performance on actual, real JD texts. Nevertheless, we have human-annotated evaluation data. Moreover, in our evaluation process, we use only one gold-standard ESCO title as the correct answer. This approach may not adequately represent a real-world scenario, where multiple ESCO titles could be correct as shown in [Table 10.3](#).

---

**Algorithm 5:** Find the most similar n-gram to a target subspan
 

---

**Data:** *sentence*: The input sentence  
*target\_subspan*: The target subspan  
*threshold*: The Levenshtein distance similarity threshold  
**Result:** *most\_similar\_ngram*: The most similar n-gram

```

1 all_ngrams ← GenerateAllNgrams(sentence)
2 filtered_ngrams ← FilterNgrams(all_ngrams, target_subspan,
   threshold)
3 most_similar_ngram ← None
4 max_similarity ← 0
5 for ngram in filtered_ngrams do
6   subspan_embedding ← EncodeWithSBERT(target_subspan)
7   ngram_embedding ← EncodeWithSBERT(ngram)
8   similarity ← CosineSimilarity(subspan_embedding,
   ngram_embedding)
9   if similarity > max_similarity and similarity > 0.5 then
10    max_similarity ← similarity
11    most_similar_ngram ← ngram
12  else
13    most_similar_ngram = UNK
14 return most_similar_ngram

```

---

## 10.6 Appendix

### 10.6.1 Data Preprocessing

We outline the preprocessing steps for the training set. In [Decorte et al. \(2023\)](#), there are sentence–ESCO skill title pairs. The data is synthetically generated by GPT-3.5. Where for each ESCO skill title a set of 10 sentences is generated. A crucial limitation for entity linkers is that the generated sentence does not have the ESCO skill title as an exact match in the sentence, but at most slightly paraphrased. To find the most similar subspan in the sentence to the target skill, we have to apply some heuristics. In [Algorithm 5](#), we denote our algorithm to find the most similar subspan. Our method is a brute force approach, where we create all possible n-grams

until the maximum length of the sentence, and compare the target subspan against each n-gram. Based on Levenshtein distance, we filter the results, where we only take the top 80% n-grams. Then, we encode both target subspan and n-gram with SentenceBERT (Reimers and Gurevych, 2019), the similarity is based on cosine similarity. If the similarity does not exceed 0.5, the candidate subspan is UNK and the ESCO title will also be UNK, otherwise, we take the most similar n-gram. Empirically, we found that these thresholds worked best. Note that this method is not error-prone, but allows us to generate implicit and negative examples to train entity linkers. We show two qualitative examples in Figure 10.1 and discuss the quality in Subsection 10.6.2.

	Train Source	Acc@1	Acc@4	Acc@8	Acc@16	Acc@32
BLINK (bert-large) UNK	Wiki + ESCO	1.38 $\pm$ 0.12	3.32 $\pm$ 0.22	4.67 $\pm$ 0.33	7.68 $\pm$ 0.42	10.70 $\pm$ 0.58
GENRE (bart-large) UNK	Wiki + ESCO	1.65 $\pm$ 0.20	4.99 $\pm$ 0.50	9.23 $\pm$ 0.58	16.01 $\pm$ 0.48	24.70 $\pm$ 2.52

Table 10.4: UNK **Linking Results**. We show the results of BLINK and GENRE predicting UNK. We use the best-performing models, based on Table 10.2.

## 10.6.2 Data Examples

We show a couple of data examples from the training (Figure 10.1) and development set (Figure 10.2). In the training examples, we show an example with a mention that is the same as the original ESCO title (“young horse training”). In addition, we have an example where there is an “implicit” mention (i.e., the mention does not exactly match with the label title). This shows that our algorithm works to an extent. For the development example, this is another implicit mention. However, these samples are human annotated. There are also quite some UNKs given the training data. We show that this is helping the model predict UNK.

### 10.6.3 Implementation Details

For training both BLINK<sup>9</sup> and GENRE,<sup>10</sup> we use their respective repositories. All models are trained for 10 epochs, for a batch size of 32 for training and 8 for evaluation. For both BLINK and GENRE we use 5% warmup. For the base models we use learning rate  $2 \times 10^{-5}$  and for the large models we use  $2 \times 10^{-6}$ . The maximum context and candidate length is 128 for both models. Each model is trained on an NVIDIA A100 GPU with 40GBs of VRAM and an AMD Epyc 7662 CPU. The seed numbers the models are initialized with are 276800, 381552, 497646, 624189, 884832. We run all models with the maximum number of epochs (10) and select the best-performing one based on validation set performance for accuracy@1.

### 10.6.4 UNK Evaluation

In Table 10.4, we show the performance of both BLINK and GENRE on the UNK label. We use the best-performing models based on Table 10.2. Generally, we observe that GENRE is better in predicting UNKs than BLINK. However, the exact linking results (i.e., Acc@1) are low. This can potentially be alleviated by actively training for predicting UNKs (Zhu et al., 2023b).

---

<sup>9</sup><https://github.com/facebookresearch/BLINK>

<sup>10</sup><https://github.com/facebookresearch/genre>

```
1  {
2    "context_left": "we're looking for someone who is
3    passionate about",
4    "context_right": "and eager to share their
5    knowledge with others.",
6    "mention": "young horse training",
7    "label_title": "young horses training",
8    "label": "Principles & techniques of educating
9    young horses important simple body control
10   exercises.",
11   "label_id": 2198
12 }
13 {
14   "context_left": "Hands-on experience with",
15   "context_right": "is a must-have qualification
16   for this job.",
17   "mention": "various hand-operated printing
18   devices",
19   "label_title": "types of hand-operated printing
20   devices",
21   "label": "Process of creating various types
22   hand-operated printing devices, such as stamps,
23   seals, embossing labels or inked pads and their
24   applications.",
25   "label_id": 10972
26 }
```

Figure 10.1: **Two Training Examples.** The training examples are in the format for BLINK, there is the left context, right context, and the mention. The label title is the ESCO skill, and the label is the description of the label title. The label ID is the ID that refers to the label title.

```
1  {  
2    "context_left": "You must have an",  
3    "context_right": "with a high-quality mindset.",  
4    "mention": "analytical proactive and structured  
5    workstyle",  
6    "label_title": "work in an organised manner",  
7    "label": "Stay focused on the project at hand,  
8    at any time. Organise, manage time, plan,  
9    schedule and meet deadlines.",  
10   "label_id": 3884  
11 }
```

Figure 10.2: **One Evaluation Example.** The evaluation example is in the format for BLINK, there is the left context, right context, and the mention. The label title is the ESCO skill, and the label is the description of the label title. The label ID is the ID that refers to the label title.

**Part V**

**Conclusions**





# Chapter 11

## Summary and Conclusions

The landscape of recent technological advances is transforming online talent platforms, including job boards and social media. Job seekers and employers are increasingly leveraging digital tools to post and apply for job opportunities, as well as to discover or provide training opportunities (Spöttl and Windelband, 2021). To effectively manage talent and bridge skills gaps in education and training programs, companies, education institutions, and training providers require up-to-date and transparent information on skills and qualifications, which can be derived from large-scale job vacancy data.

In this thesis, we introduce *Computational Job Market Analysis* and make contributions to three areas in this domain: 1) Annotation, datasets, and domain-specific models (i.e., resources) in [Chapter 3](#), [Chapter 4](#), [Chapter 5](#), and [Chapter 6 \(Part II\)](#). 2) Several modeling methodologies for skill extraction based on domain adaptation, weak supervision, and retrieval-augmentation in [Chapter 7](#), [Chapter 8](#), and [Chapter 9 \(Part III\)](#). 3) Last, exploring how to link extracted spans to an existing taxonomy in [Chapter 10 \(Part IV\)](#). In the following part, we conclude the thesis with answers to the proposed research questions ([Chapter 1](#)) and suggestions for future work ([Section 11.1](#)).

**RQ1: How do Transformer-based models compare against recurrent neural network models for de-identification of privacy-related entities in job postings?**

In [Chapter 3](#), we introduced JOBSTACK, a dataset for de-identification of privacy-related entities in job postings. We posed this task as a sequence labeling task and used both BiLSTM-based ([Plank et al., 2016](#)) and transformer-based models ([Devlin et al., 2019b](#)) and compared them to each other on this task. We found that transformer-based models consistently outperform BiLSTM-based models that have been standard for de-identification in the medical domain, around 6% span-F1 on the development set and 1% span-F1 on the test set of JOBSTACK.

**RQ2: Which auxiliary tasks and datasets improve de-identification performance of privacy-related entities from job postings?**

For the task of de-identification, we explored multi-task learning, where we made use of varied auxiliary data: Generic named entity recognition (CoNLL2003; [Tjong Kim Sang, 2002b](#)) and de-identification in medical data (I2B2/UTHealth; [Stubbs and Uzuner, 2015b](#)) to improve upon the task of de-identification of job postings. We showed that, indeed, making use of a close downstream task was beneficial, where I2B2/UTHealth was the most helpful for overall span-F1 performance. In our analysis, we found that I2B2/UTHealth is particularly helpful for predicting profession entities and with CoNLL2003, the model predicts locations slightly better. A combination of both CoNLL2003 and I2B2/UTHealth resulted in the highest recall scores on the test set. This shows the usefulness of multi-task learning for this task ([Chapter 3](#)).

**RQ3: Can training dynamic-based active learning lead to more efficient model training than previous methods for text classification while maintaining performance?**

In [Chapter 4](#), we proposed a novel active learning algorithm: Cartography Active Learning, as an example for a speed up of annotation and more efficient model training. We optimize for selecting the most informative data concerning a model by leveraging insights from data maps ([Swayamdipta et al., 2020a](#)). We show empirically that our method is on par or significantly outperforms various popular active learning methods based on uncertainty sampling ([Lewis and Gale, 1994](#); [Lewis and Catlett, 1994](#)) or diverse batching ([Geifman and El-Yaniv, 2017](#); [Sener and Savarese, 2018](#); [Gissin and Shalev-Shwartz, 2019](#); [Zhdanov, 2019](#)). We show in our analysis that our algorithm chooses more informative samples, has less overlap of instances with other active learning algorithms, and performs equally or better than full-dataset performance using <14% training data.

**RQ4: What are skills and how challenging is it to manually identify them in job postings?**

In [Chapter 5](#), we start by developing annotation guidelines for skill extraction. We draw inspiration from the ESCO taxonomy and annotate for skill and knowledge components on the token level with three annotators. We define several annotation rules for knowledge and skill annotation, such as: “knowledge is something that one possesses, and cannot (usually) physically execute”. For example, knowledge of the “Python” programming language. For a skill, we expect it to start with a verb, but if it includes a modal verb, the modal verb will not be tagged. For instance, in the sentence “you will train new staff”, the skill will be “train new staff”. After six rounds of annotation, we calculated the agreement over tokens and the span surface form. We obtain a Fleiss’  $\kappa$  of 0.70–0.75, which indicates a substantial agreement (i.e., accuracy) over the span annotations. In the end, we introduce SKILLSPAN, a job posting dataset of more than 10,000 sentences annotated for skill and knowledge components.

**RQ5: To what extent does continuous pre-training on unlabeled job posting data improve skill extraction in English?**

We release domain-adapted BERT and SpanBERT (Joshi et al., 2020b) models—JobBERT and JobSpanBERT. In this scenario, domain-adapted means further training of the language model using the pre-training objectives they have been trained on (e.g., masked language modeling). We train BERT and SpanBERT further on 3.2M English job posting sentences. Our analysis shows that domain-adaptive pre-training helps to improve downstream performance on the task of skill extraction by around 2 span-F1 on the SKILLSPAN test set.

**RQ6: How much does continuous pre-training on large-scale unlabeled Danish job posting data improve skill classification in Danish?**

We showed that domain adaptive pre-training helps improve performance on the task of skill extraction for English and expect it to work as well on a mid-resource language like Danish. We contribute DAJOBBERT, our domain-adapted language model for the Danish job market (Chapter 6) continuously pre-trained on 24.5M Danish job posting sentences. It improves upon regular Danish BERT (DaBERT) by 19.4% weighted macro-F1 on the task of skill classification.

**RQ7: How effective is distant supervision with ESCO for skill categorization?**

Additionally, in Chapter 6, we present a novel skill classification dataset for competences in Danish: KOMPETENCER. We made use of the ESCO taxonomy (le Vrang et al., 2014) to categorize extracted spans in Danish job postings into their respective taxonomy code using distant supervision (defined in Subsection 2.4.3). Via manual inspection, we found that 70.4% of distantly supervised labels were correct for the Danish development split and 14.1% correct for the Danish test split. In the test split, there were frequently unknown spans. Finally, we manually correct the distantly supervised labels.

**RQ8: Can we apply existing skill labels from the ESCO taxonomy as a weak supervision signal for skill extraction?**

In [Chapter 7](#), we explore the compatibility of the ESCO skill taxonomy as a weak supervision signal for skill extraction. Employing various skill representation methods established in prior research, we demonstrate the usefulness of ESCO skill representations in this context. Our results show mostly gains in loose-F1 score ([Section 7.3](#)), suggesting a substantial overlap between predicted and gold spans. However, there is a need for refinement in offset methods to precisely extract the correct span, which can be achieved through techniques like human post-editing or automated methods such as candidate filtering.

**RQ9: Can we use a graph-guided pre-training method with ESCO to improve a multilingual language model’s performance on skill extraction and classification?**

In [Chapter 8](#), we introduce ESCOXML-R: A domain-adapted, multilingual language model further pre-trained on the ESCO taxonomy via a link-aware pre-training approach ([Yasunaga et al., 2022b](#)). We evaluated across an extensive set of datasets in the domain across four languages, ESCOXML-R outperformed XLM-R<sub>large</sub> in six out of nine job-related tasks, consisting of skill extraction and classification. We attribute the enhanced performance of ESCOXML-R to its effectiveness on shorter span lengths, emphasizing the value of pre-training on the ESCO dataset. Notably, ESCOXML-R exhibited improved performance on both frequent surface spans and different span lengths.

**RQ10: How does a retrieval-augmented language model as a dataset-unifying method affect robustness for skill extraction?**

Skills can be mentioned in different frequencies across multiple datasets. In [Chapter 9](#), we showed a dataset-unifying method for combining several

skill extraction datasets to address the challenge of the long-tail distribution of skills (Subsection 2.1.2). We present NNOSE, an LM that integrates a non-parametric datastore for nearest neighbor retrieval of skill tokens of three combined datasets for skill extraction. NNOSE was assessed on three diverse skill datasets, showcasing its ability to enhance the performance of all language models used without additional parameter tuning with a significant performance increase on the in-dataset setting. Our analysis indicates that NNOSE effectively leverages all datasets by retrieving tokens from each, demonstrating proficiency not only with rare, but also improving the performance of more common skills. Notably, models' performance without an external datastore deteriorates in cross-dataset settings, but the introduction of NNOSE leads to overall improvement across all cross-dataset scenarios, with the largest improvement of 30.7% span-F1. These findings suggest that NNOSE holds promise for application-specific skill extraction setups and further qualitative analysis may uncover skills overlooked in manual annotations in other datasets.

**RQ11: How effective are entity linking methodologies for linking skills to knowledge bases (e.g., ESCO)?**

Lastly, in Chapter 10, we explore entity linking in the job market domain to finally match the extracted skills with a taxonomical counterpart instead of multi-class classification as used in Chapter 6. We used two established neural models. Our findings indicate that the bi-encoder architecture of BLINK outperforms the autoregressive GENRE model in strict linking results. Strict linking performance is modest due to evaluation set limitations, considering only one correct skill per mention. However, upon closer examination of predictions, valid links emerge, suggesting the potential for multiple correct links per mention. This underscores the need for a more comprehensive evaluation approach.

## 11.1 Future Directions

**End-to-End Datasets** As categorizing skills can give us a snapshot of the current labor market, it is important to develop capable language technology systems that can do this succinctly. Strictly speaking, [Chapter 10](#) shows the most comprehensive way to do the categorization. However, this is not the only approach, as one could directly go from sentence or skill to taxonomy title as shown in [Chapter 6](#) and ([Decorte et al., 2023](#); [Clavié and Soulié, 2023](#)). A possible approach to building a large-scale dataset for matching skills with taxonomy codes in the most efficient way is to consider it as a multi-label problem: One sentence could have multiple possible skill codes. This not only allows for training a model to do this task but also to evaluate fairly, where multiple taxonomy codes are valid predictions.

If there is a scenario where it is interesting to see the matched span in a sentence with a grounded skill, one can also decide to still think of the task as an entity linking problem, with multiple valid skills.

**Large Language Models** Assuming that we want to extract skill spans from the sentences or documents, another challenge is discontinuous spans as indicated in [Chapter 2](#). Given the skill “*critical, independent, and open-minded thinking skills*”, one has to make modeling choices here: Are we keeping this as a *multiple skill span* or do we separate this conjunction in “critical thinking skills”, “independent thinking skills”, “open-minded thinking skills”? One hypothesis is that generative large language models can solve this task, by automatically linking the skill span at hand to a taxonomy entry. In addition, there needs to be some effort on how the gold labels are represented and to evaluate this correctly.

We are currently in exciting times, the capabilities of large language models are showing signs of general task solvers ([McCann et al., 2018](#); [Sanh et al., 2021](#); [Wei et al., 2022](#); [Muennighoff et al., 2022](#)). In the context of this thesis, one interesting direction is to investigate to what extent these systems can do end-to-end extraction and matching. Now,

in these models, context length is not a prevailing issue anymore. Thus, full job descriptions can be used as input to the model. As the output is generated, one needs to consider constraining it to a specific subset of data (e.g., ESCO).

On a broader note, skill extraction is one of the many applications within Computational Job Market Analysis. Skills can be extracted from resumes and other types of documents. One can use the extracted skills as features for recommending jobs or courses to a specific individual or other types of talent acquisition. Additionally, we can create systems for career growth guidance. Concurrently, the integration of automated methodologies in such applications introduces potential challenges and uncertainties related to fairness, privacy, reproducibility, controllability, and transparency. To address these concerns, it is evident that research in NLP plays a pivotal role, offering insights and solutions to tackle the aforementioned risks.



## Bibliography

Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010. URL <https://pubs.rsc.org/en/content/articlehtml/2014/ay/c3ay41907j>. pages 219

John Aberdeen, Samuel Bayer, Reyhan Yeniterzi, Ben Wellner, Cheryl Clark, David Hanauer, Bradley Malin, and Lynette Hirschman. The mitre identification scrubber toolkit: design, training, and assessment. *International journal of medical informatics*, 79(12):849–859, 2010. pages 44

Accountability Act. Health insurance portability and accountability act of 1996. *Public law*, 104:191, 1996. pages 30, 44

Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA, 2018. Association for Computational Linguistics. URL <https://aclanthology.org/C18-1139>. pages 45

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA, June

- 2019a. Association for Computational Linguistics. doi: 10.18653/v1/W19-1909. URL <https://aclanthology.org/W19-1909>. pages 85, 102
- Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA, 2019b. Association for Computational Linguistics. doi: 10.18653/v1/W19-1909. URL <https://aclanthology.org/W19-1909>. pages 172, 206
- Ziqiao Ao, Gergely Horváth, Chunyuan Sheng, Yifan Song, and Yutong Sun. Skill requirements in job advertisements: A comparison of skill-categorization methods based on wage regressions. *Information Processing & Management*, 60(2):103185, 2023. URL [https://www.sciencedirect.com/science/article/pii/S0306457322002862?casa\\_token=g-IWrRrn4vEAAAAA:3Qe7yyjupAwY1BgFCeIf-psXEx\\_7roe-kXZi36buA0BVZ6WZfCcJgkyP0pUWAatCL7upHSPz2HXV7](https://www.sciencedirect.com/science/article/pii/S0306457322002862?casa_token=g-IWrRrn4vEAAAAA:3Qe7yyjupAwY1BgFCeIf-psXEx_7roe-kXZi36buA0BVZ6WZfCcJgkyP0pUWAatCL7upHSPz2HXV7). pages 213
- Adam Mehdi Arafan, David Graus, Fernando P Santos, and Emma Beauxis-Aussalet. End-to-end bias mitigation in candidate recommender systems with fairness gates. 2022. pages 185, 216
- Andrew Arnold, Ramesh Nallapati, and William W Cohen. A comparative study of methods for transductive transfer learning. In *Seventh IEEE international conference on data mining workshops (ICDMW 2007)*, pages 77–82. IEEE, 2007. pages 37
- Mikel Artetxe, Sebastian Ruder, Dani Yogatama, Gorka Labaka, and Eneko Agirre. A call for more rigor in unsupervised cross-lingual learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7375–7388, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.658. URL <https://aclanthology.org/2020.acl-main.658>. pages 134

David H Autor and David Dorn. The growth of low-skill service jobs and the polarization of the us labor market. *American economic review*, 103(5):1553–1597, 2013. URL <https://www.aeaweb.org/articles?id=10.1257/aer.103.5.1553>. pages 28, 198

David H Autor, Frank Levy, and Richard J Murnane. The skill content of recent technological change: An empirical exploration. *The Quarterly journal of economics*, 118(4):1279–1333, 2003. URL <https://academic.oup.com/qje/article-abstract/118/4/1279/1925105?login=false>. pages 28, 198

Tom Ayoola, Shubhi Tyagi, Joseph Fisher, Christos Christodoulopoulos, and Andrea Pierleoni. ReFinED: An efficient zero-shot-capable approach to end-to-end entity linking. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, pages 209–220, Hybrid: Seattle, Washington + Online, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-industry.24. URL <https://aclanthology.org/2022.naacl-industry.24>. pages 9, 233

José Azar, Ioana Marinescu, and Marshall Steinbaum. Labor market concentration. *Journal of Human Resources*, 57(S):S167–S199, 2022. pages 2

Dominic Balasuriya, Nicky Ringland, Joel Nothman, Tara Murphy, and James R. Curran. Named entity recognition in Wikipedia. In *Proceedings of the 2009 Workshop on The People’s Web Meets NLP: Collaboratively Constructed Semantic Resources (People’s Web)*, pages 10–18, Suntec, Singapore, 2009. Association for Computational Linguistics. URL <https://aclanthology.org/W09-3302>. pages 51

Krisztian Balog, Yi Fang, Maarten De Rijke, Pavel Serdyukov, and Luo Si. Expertise retrieval. *Foundations and Trends in Information Retrieval*, 6(2–3):127–256, 2012. pages 2, 84, 124, 152, 166, 198, 232

- Carmen Banea, Di Chen, Rada Mihalcea, Claire Cardie, and Janyce Wiebe. SimCompass: Using deep learning word embeddings to assess cross-level similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 560–565, Dublin, Ireland, August 2014. Association for Computational Linguistics. doi: 10.3115/v1/S14-2098. URL <https://aclanthology.org/S14-2098>. pages 80
- Maria Barrett, Max Müller-Eberstein, Elisa Bassignana, Amalie Brogaard Pauli, Mike Zhang, and Rob van der Goot. Can humans identify domains?, 2024. pages
- Elisa Bassignana, Max Müller-Eberstein, Mike Zhang, and Barbara Plank. Evidence > intuition: Transferability estimation for encoder selection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4218–4227, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL <https://aclanthology.org/2022.emnlp-main.283>. pages
- David Beauchemin, Julien Laumonier, Yvan Le Ster, and Marouane Yassine. “FIJO”: a french insurance soft skill detection dataset. *arXiv e-prints*, pages arXiv–2204, 2022. pages xvi, 166, 167, 177, 213
- Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1371. URL <https://aclanthology.org/D19-1371>. pages 45
- Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *ArXiv preprint*, abs/2004.05150, 2020. URL <https://arxiv.org/abs/2004.05150>. pages 84, 85, 102
- Emily M. Bender and Batya Friedman. Data statements for natural language processing: Toward mitigating system bias and enabling better science.

*Transactions of the Association for Computational Linguistics*, 6:587–604, 2018. doi: 10.1162/tacl\_a\_00041. URL <https://aclanthology.org/Q18-1041>. pages 92, 107, 141

Darina Benikova, Chris Biemann, and Marc Reznicek. NoSta-D named entity annotation for German: Guidelines and dataset. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2524–2531, Reykjavik, Iceland, 2014. European Language Resources Association (ELRA). URL [http://www.lrec-conf.org/proceedings/lrec2014/pdf/276\\_Paper.pdf](http://www.lrec-conf.org/proceedings/lrec2014/pdf/276_Paper.pdf). pages 99

Janek Bevendorff, Khalid Al Khatib, Martin Potthast, and Benno Stein. Crawling and preprocessing mailing lists at scale for dialog analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1151–1158, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.108. URL <https://aclanthology.org/2020.acl-main.108>. pages 45

Rishabh Bhardwaj, George Polovets, and Monica Sunkara. Adaptation approaches for nearest neighbor language models. *ArXiv preprint*, abs/2211.07828, 2022. URL <https://arxiv.org/abs/2211.07828>. pages 214

Akshay Bhola, Kishaloy Halder, Animesh Prasad, and Min-Yen Kan. Retrieving skills from job descriptions: A language model based extreme multi-label classification framework. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5832–5842, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.513. URL <https://aclanthology.org/2020.coling-main.513>. pages xvi, 25, 84, 87, 88, 139, 152, 166, 176, 213

John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment

- classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL <https://aclanthology.org/P07-1056>. pages 37
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. doi: 10.1162/tacl\_a\_00051. URL <https://aclanthology.org/Q17-1010>. pages 80
- Carlo Bonferroni. Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze*, 8:3–62, 1936. pages 75, 101, 120, 143, 146, 147
- Roberto Boselli, Mirko Cesarini, Fabio Mercorio, and Mario Mezzanzanica. Classifying online job advertisements through machine learning. *Future Generation Computer Systems*, 86:319–328, 2018. pages 139
- Dmitriy Botov, Julius Klenin, Andrey Melnikov, Yuri Dmitrin, Ivan Nikolaev, and Mikhail Vinel. Mining labor market requirements using distributional semantic models and deep learning. In Witold Abramowicz and Rafael Corchuelo, editors, *Business Information Systems - 22nd International Conference, BIS 2019, Seville, Spain, June 26-28, 2019, Proceedings, Part II*, volume 354 of *Lecture Notes in Business Information Processing*, pages 177–190. Springer, 2019. doi: 10.1007/978-3-030-20482-2\_15. URL [https://doi.org/10.1007/978-3-030-20482-2\\_15](https://doi.org/10.1007/978-3-030-20482-2_15). pages 6
- Erik Brynjolfsson and Andrew McAfee. *Race against the machine: How the digital revolution is accelerating innovation, driving productivity, and irreversibly transforming employment and the economy*. Brynjolfsson and McAfee, 2011. pages 1, 152, 166, 198, 232
- Erik Brynjolfsson and Andrew McAfee. *The second machine age: Work, progress, and prosperity in a time of brilliant technologies*. WW Norton & Company, 2014. URL <https://books.google.nl/books?hl=nl&lr>

=&id=WiKwAgAAQBAJ&oi=fnd&pg=PA1&dq=The+second+machine+age:+  
+Work,+progress,+and+prosperity+in+a+time+of+brilliant+techn  
ologies&ots=4\_-uUc0Acg&sig=rh-Nl7fDit4mmdb\_yMiATI9MkZA. pages  
1, 152, 166, 198, 232

Marlis Buchmann, Helen Buchs, Felix Busch, Simon Clematide, Ann-Sophie Gnehm, and Jan Müller. Swiss job market monitor: A rich source of demand-side micro data of the labour market. *European Sociological Review*, 2022. pages 177

Iacer Calixto, Alessandro Raganato, and Tommaso Pasini. Wikipedia entities as rendezvous across languages: Grounding multilingual language models by predicting Wikipedia hyperlinks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3651–3661, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.286. URL <https://aclanthology.org/2021.naacl-main.286>. pages 182

de Nicola Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. Autoregressive entity retrieval. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=5k8F6UU39V>. pages 9, 232, 233, 236, 237

Lina Cao and Jian Zhang. Skill requirements analysis for data analysts based on named entities recognition. In *2021 2nd International Conference on Big Data and Informatization Education (ICBDIE)*, pages 64–68, 2021. doi: 10.1109/ICBDIE52740.2021.00023. pages 6

Nicola de Cao, Ledell Wu, Kashyap Popat, Mikel Artetxe, Naman Goyal, Mikhail Plekhanov, Luke Zettlemoyer, Nicola Cancedda, Sebastian Riedel, and Fabio Petroni. Multilingual autoregressive entity linking. *Transactions of the Association for Computational Linguistics*, 10:274–290, 2022. pages 9, 233

- Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 267–284, 2019. pages 42
- Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, and Arie Herbert-Voss. Extracting training data from large language models. *ArXiv preprint*, abs/2012.07805, 2020. URL <https://arxiv.org/abs/2012.07805>. pages 42
- Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997. pages 3, 35, 52, 84, 85, 98
- Branden Chan, Stefan Schweter, and Timo Möller. German’s next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online), 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.598. URL <https://aclanthology.org/2020.coling-main.598>. pages 177
- Tao Chen and Min-Yen Kan. Creating a live, public short message service corpus: the nus sms corpus. *Language Resources and Evaluation*, 47(2): 299–335, 2013. pages 3, 46
- Mariia Chernova. Occupational skills extraction with FinBERT. *Master’s Thesis*, 2020. URL [https://www.theseus.fi/bitstream/handle/10024/348657/Mariia\\_Chernova\\_Master\\_Thesis\\_full.pdf?sequence=2](https://www.theseus.fi/bitstream/handle/10024/348657/Mariia_Chernova_Master_Thesis_full.pdf?sequence=2). pages 87, 89, 152, 166, 213
- Hyung Won Chung, Thibault Févry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. Rethinking embedding coupling in pre-trained language models. *ArXiv preprint*, abs/2010.12821, 2020. URL <https://arxiv.org/abs/2010.12821>. pages 124, 125, 133, 175
- Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D. Manning, and Quoc V. Le. BAM! born-again multi-task networks for nat-



- ural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5931–5937, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1595. URL <https://aclanthology.org/P19-1595>. pages 57
- Benjamin Clavié and Guillaume Soulié. Large language models as batteries-included zero-shot esco skills matchers. *ArXiv preprint*, abs/2307.03539, 2023. URL <https://arxiv.org/abs/2307.03539>. pages 10, 233, 255
- David Cohn, Zoubin Ghahramani, and Michael Jordan. Active learning with statistical models. *Advances in neural information processing systems*, 7, 1994. pages 3, 31
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.747. URL <https://aclanthology.org/2020.acl-main.747>. pages 7, 169
- Mark Craven, Johan Kumlien, et al. Constructing biological knowledge bases by extracting information from text sources. In *ISMB*, volume 1999, pages 77–86, 1999. pages 32
- Aron Culotta and Andrew McCallum. Reducing labeling effort for structured prediction tasks. In *AAAI*, volume 5, pages 746–751, 2005. pages 64, 71
- Ido Dagan and Sean P Engelson. Committee-based sampling for training probabilistic classifiers. In *Machine Learning Proceedings 1995*, pages 150–157. Elsevier, 1995. pages 64, 72

- Jens-Joris Decorte, Jeroen Van Hautte, Thomas Demeester, and Chris Develder. Jobbert: Understanding job titles through skills. *ArXiv preprint*, abs/2109.09605, 2021. URL <https://arxiv.org/abs/2109.09605>. pages 167
- Jens-Joris Decorte, Jeroen Van Hautte, Johannes Deleu, Chris Develder, and Demeester. Design of negative sampling strategies for distantly supervised skill extraction. *ArXiv preprint*, abs/2209.05987, 2022. URL <https://arxiv.org/abs/2209.05987>. pages 10, 166, 213, 233, 234, 235
- Jens-Joris Decorte, Severine Verlinden, Jeroen Van Hautte, Johannes Deleu, Chris Develder, and Thomas Demeester. Extreme multi-label skill extraction training using large language models. *ArXiv preprint*, abs/2307.10778, 2023. URL <https://arxiv.org/abs/2307.10778>. pages 10, 233, 234, 235, 242, 255
- Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4418. URL <https://aclanthology.org/W17-4418>. pages 181
- Franck Dernoncourt, Ji Young Lee, Ozlem Uzuner, and Peter Szolovits. De-identification of patient notes with recurrent neural networks. *Journal of the American Medical Informatics Association*, 24(3):596–606, 2017. pages 44, 45
- Ketki V Deshpande, Shimei Pan, and James R Foulds. Mitigating demographic bias in ai-based resume filtering. In *Adjunct publication of the 28th ACM conference on user modeling, adaptation and personalization*, pages 268–275, 2020. pages 185, 216
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language under-

- standing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019a. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL <https://aclanthology.org/N19-1423>. pages 235, 237
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, 2019b. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL <https://aclanthology.org/N19-1423>. pages xiii, 45, 52, 84, 88, 97, 125, 133, 155, 169, 202, 250
- Rotem Dror, Segev Shlomov, and Roi Reichart. Deep dominance - how to properly compare deep neural models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2773–2785, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1266. URL <https://aclanthology.org/P19-1266>. pages 74, 101, 117, 143, 146
- Yichao Du, Zhirui Zhang, Bingzhe Wu, Lema Liu, Tong Xu, and Enhong Chen. Federated Nearest Neighbor Machine Translation. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=R1U5G2spbLd>. pages 214
- Elisabeth Eder, Ulrike Krieg-Holz, and Udo Hahn. De-identification of emails: Pseudonymizing privacy-sensitive data in a German email corpus. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 259–269, Varna, Bulgaria, 2019. INCOMA Ltd. doi: 10.26615/978-954-452-056-4\_030. URL <https://aclanthology.org/R19-1030>. pages 45, 47

- Liat Ein-Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim. Active Learning for BERT: An Empirical Study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7949–7962, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.638. URL <https://aclanthology.org/2020.emnlp-main.638>. pages 63, 66, 70
- Peter Elias. Occupational classification (isco-88): Concepts, methods, reliability, validity and cross-national comparability. Technical report, OECD Publishing, 1997. pages 20, 89, 139
- Tyna Eloundou, Sam Manning, Pamela Mishkin, and Daniel Rock. Gpts are gpts: An early look at the labor market impact potential of large language models. *ArXiv preprint*, abs/2303.10130, 2023. URL <https://arxiv.org/abs/2303.10130>. pages 1
- ESCO. Machine Learning Assisted Mapping of Multilingual Occupational Data to ESCO (Part 1), 2022. URL <https://esco.ec.europa.eu/en/about-esco/data-science-and-esco/machine-learning-assisted-mapping-multilingual-occupational-data-esco-part-1>. pages 167
- Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1006. URL <https://aclanthology.org/D19-1006>. pages 202
- European Commission. Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free

- movement of such data, and repealing directive 95/46. *Official Journal of the European Union (OJ)*, 59(1-88):294, 2016. pages 2, 10, 30, 44, 47
- European Commission. Industry 5.0: Towards more sustainable, resilient and human-centric industry, 2021. URL [https://research-and-innovation.ec.europa.eu/news/all-research-and-innovation-news/industry-50-towards-more-sustainable-resilient-and-human-centric-industry-2021-01-07\\_en](https://research-and-innovation.ec.europa.eu/news/all-research-and-innovation-news/industry-50-towards-more-sustainable-resilient-and-human-centric-industry-2021-01-07_en). pages 1
- Silvia Fareri, Nicola Melluso, Filippo Chiarello, and Gualtiero Fantoni. Skillner: Mining and mapping soft skills from any text. *Expert Systems with Applications*, 184:115544, 2021. URL [https://www.sciencedirect.com/science/article/pii/S0957417421009519?casa\\_token=r5CvNzj74-gAAAAA:CHR3Dmf0ze1nTt359q7WFNNHPSJhNUVYZ5qCxcZS-\\_a9e1s3VIHLkGTGkwi745\\_Rsn74za-BYsvE](https://www.sciencedirect.com/science/article/pii/S0957417421009519?casa_token=r5CvNzj74-gAAAAA:CHR3Dmf0ze1nTt359q7WFNNHPSJhNUVYZ5qCxcZS-_a9e1s3VIHLkGTGkwi745_Rsn74za-BYsvE). pages 213
- Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971. pages 50, 94, 130
- Joseph L Fleiss and Jacob Cohen. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and psychological measurement*, 33(3):613–619, 1973. pages 50, 94, 130
- Max Friedrich, Arne Köhn, Gregor Wiedemann, and Chris Biemann. Adversarial learning of privacy-preserving text representations for de-identification of medical records. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5829–5839, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1584. URL <https://aclanthology.org/P19-1584>. pages 3, 42, 43, 45
- Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the*

- 33rd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1050–1059. JMLR.org, 2016. URL <http://proceedings.mlr.press/v48/gal16.html>. pages 4, 62, 64, 72
- Octavian-Eugen Ganea and Thomas Hofmann. Deep joint entity disambiguation with local neural attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2619–2629, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1277. URL <https://aclanthology.org/D17-1277>. pages 234
- Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. Representation degeneration problem in training natural language generation models. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL <https://openreview.net/forum?id=SkEYojRqtm>. pages 202
- Adrian Gardiner, Cheryl Aasheim, Paige Rutner, and Susan Williams. Skill requirements in big data: A content analysis of job advertisements. *Journal of Computer Information Systems*, 58(4):374–384, 2018. pages 139
- Yonatan Geifman and Ran El-Yaniv. Deep active learning over the long tail. *ArXiv preprint*, abs/1711.00941, 2017. URL <https://arxiv.org/abs/1711.00941>. pages 4, 62, 66, 251
- Anna Giabelli, Lorenzo Malandri, Fabio Mercurio, and Mario Mezzanzanica. Graphlmi: A data driven system for exploring labor market information through graph databases. *Multimedia Tools and Applications*, pages 1–30, 2020. pages 139
- Daniel Gissin and Shai Shalev-Shwartz. Discriminative active learning. *ArXiv preprint*, abs/1907.06347, 2019. URL <https://arxiv.org/abs/1907.06347>. pages 4, 62, 65, 66, 70, 72, 79, 251

- Ann-Sophie Gnehm and Simon Clematide. Text zoning and classification for job advertisements in German, French and English. In *Proceedings of the Fourth Workshop on Natural Language Processing and Computational Social Science*, pages 83–93, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.nlpccs-1.10. URL <https://aclanthology.org/2020.nlpccs-1.10>. pages 177
- Ann-sophie Gnehm, Eva Bühlmann, Helen Buchs, and Simon Clematide. Fine-grained extraction and classification of skill requirements in German-speaking job ads. In *Proceedings of the Fifth Workshop on Natural Language Processing and Computational Social Science (NLP+CSS)*, pages 14–24, Abu Dhabi, UAE, November 2022a. Association for Computational Linguistics. URL <https://aclanthology.org/2022.nlpccs-1.2>. pages 9, 233
- Ann-Sophie Gnehm, Eva Bühlmann, and Simon Clematide. Evaluation of transfer learning and domain adaptation for analyzing german-speaking job advertisements. In *Proceedings of the Language Resources and Evaluation Conference*, pages 3892–3901, Marseille, France, 2022b. European Language Resources Association. URL <https://aclanthology.org/2022.lrec-1.414>. pages xvi, 152, 166, 167, 177, 213
- Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. *Linear algebra*, 2:134–151, 1971. pages 218
- Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. pages 1
- Nidhi Goyal, Jushaan Kalra, Charu Sharma, Raghava Mutharaju, Niharika Sachdeva, and Ponnurangam Kumaraguru. JobXMLC: EXtreme multi-label classification of job skills with graph neural networks. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2181–2191, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. URL <https://aclanthology.org/2023.findings-eacl.163>. pages 25, 166, 213

- Edouard Grave, Armand Joulin, and Nicolas Usunier. Improving neural language models with a continuous cache. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL <https://openreview.net/forum?id=B184E5qee>. pages 8, 213
- Thomas Green, Diana Maynard, and Chenghua Lin. Development of a benchmark corpus to support entity recognition in job descriptions. In *Proceedings of the Language Resources and Evaluation Conference*, pages 1201–1208, Marseille, France, 2022. European Language Resources Association. URL <https://aclanthology.org/2022.lrec-1.128>. pages xvi, 152, 166, 167, 176, 204, 213
- Akshay Gugnani and Hemant Misra. Implicit skills extraction using document embedding and its use in job recommendation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 13286–13293. AAAI Press, 2020. URL <https://aaai.org/ojs/index.php/AAAI/article/view/7038>. pages 87, 213
- Dilip Gupta, Melissa Saul, and John Gilbertson. Evaluation of a deidentification (de-id) software engine to share pathology reports and clinical documents for research. *American journal of clinical pathology*, 121(2): 176–186, 2004. pages 44
- Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don’t stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online, July 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.740. URL <https://aclanthology.org/2020.acl-main.740>. pages 84



Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online, 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.740. URL <https://aclanthology.org/2020.acl-main.740>. pages 5, 85, 86, 97, 102, 125, 133, 167, 172, 204, 206, 220

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR, 2020. URL <http://proceedings.mlr.press/v119/guu20a.html>. pages 8, 213

Xiaochuang Han and Jacob Eisenstein. Unsupervised domain adaptation of contextualized embeddings for sequence labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4238–4248, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1433. URL <https://aclanthology.org/D19-1433>. pages 84, 85, 125, 172, 206

Bin He, Yi Guan, Jianyi Cheng, Keting Cen, and Wenlan Hua. Crfs based de-identification of medical records. *Journal of biomedical informatics*, 58:S39–S46, 2015. pages 44

Bin He, Di Zhou, Jinghui Xiao, Xin Jiang, Qun Liu, Nicholas Jing Yuan, and Tong Xu. BERT-MK: Integrating graph contextualized knowledge into pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2281–2290, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findi

- ngs-emnlp.207. URL <https://aclanthology.org/2020.findings-emnlp.207>. pages 7, 182
- Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. Efficient nearest neighbor language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5703–5714, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.461. URL <https://aclanthology.org/2021.emnlp-main.461>. pages 214
- Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang. Learning entity representation for entity disambiguation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 30–34, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL <https://aclanthology.org/P13-2006>. pages 9, 233
- Sepp Hochreiter, Jürgen Schmidhuber, and Corso Elvezia. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. pages 3, 88, 176
- Charles Holt and Martin David. The concept of job vacancies in a dynamic theory of the labor market. In *The measurement and interpretation of job vacancies*, pages 73–110. NBER, 1966. pages 2
- Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *ArXiv preprint*, abs/1112.5745, 2011. URL <https://arxiv.org/abs/1112.5745>. pages 4, 62, 64, 72
- Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia, 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1031. URL <https://aclanthology.org/P18-1031>. pages 125

- Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR, 2020. URL <http://proceedings.mlr.press/v119/hu20b.html>. pages 169
- Junjie Huang, Duyu Tang, Wanjun Zhong, Shuai Lu, Linjun Shou, Ming Gong, Daxin Jiang, and Nan Duan. WhiteningBERT: An easy unsupervised sentence embedding approach. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 238–244, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.23. URL <https://aclanthology.org/2021.findings-emnlp.23>. pages 202
- Gautier Izacard and Edouard Grave. Distilling knowledge from reader to retriever for question answering. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=NTEz-6wysdb>. pages 8, 213
- Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Few-shot learning with retrieval augmented language models. *ArXiv preprint*, abs/2208.03299, 2022. URL <https://arxiv.org/abs/2208.03299>. pages 8, 213
- Paul Jaccard. Distribution de la flore alpine dans le bassin des dranses et dans quelques régions voisines. *Bull Soc Vaudoise Sci Nat*, 37:241–272, 1901. pages 209
- Faizan Javed, Qinlong Luo, Matt McNair, Ferosh Jacob, Meng Zhao, and Tae Seung Kang. Carotene: A job title classification system for the online recruitment domain. In *2015 IEEE First International Conference on Big*

- Data Computing Service and Applications*, pages 286–293. IEEE, 2015. pages 167
- Faizan Javed, Matt McNair, Ferosh Jacob, and Meng Zhao. Towards a job title classification system. *ArXiv preprint*, abs/1606.00917, 2016. URL <https://arxiv.org/abs/1606.00917>. pages 167
- Faizan Javed, Phuong Hoang, Thomas Mahoney, and Matt McNair. Large-scale occupational skills normalization for online recruitment. In Sander P. Singh and Shaul Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 4627–4634. AAAI Press, 2017. URL <http://aaai.org/ocs/index.php/IAAI/IAAI17/paper/view/14922>. pages 87
- Kristian Nørgaard Jensen, Mike Zhang, and Barbara Plank. De-identification of privacy-related entities in job postings. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 210–221, Reykjavik, Iceland (Online), May 31–2 June 2021a. Linköping University Electronic Press, Sweden. URL <https://aclanthology.org/2021.nodalida-main.21>. pages 92, 98, 139
- Kristian Nørgaard Jensen, Mike Zhang, and Barbara Plank. De-identification of privacy-related entities in job postings. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 210–221, Reykjavik, Iceland (Online), 2021b. Linköping University Electronic Press, Sweden. URL <https://aclanthology.org/2021.nodalida-main.21>. pages xvi, 167, 177
- Shanshan Jia, Xiaoan Liu, Ping Zhao, Chang Liu, Lianying Sun, and Tao Peng. Representation of job-skill in artificial intelligence with knowledge graph analysis. In *2018 IEEE Symposium on Product Compliance Engineering-Asia (ISPCE-CN)*, pages 1–6. IEEE, 2018. pages 6, 87, 124
- Hui Jiang, Ziyao Lu, Fandong Meng, Chulun Zhou, Jie Zhou, Degen Huang, and Jinsong Su. Towards robust k-nearest-neighbor machine translation.

*ArXiv preprint*, abs/2210.08808, 2022a. URL <https://arxiv.org/abs/2210.08808>. pages 214

Qingnan Jiang, Mingxuan Wang, Jun Cao, Shanbo Cheng, Shujian Huang, and Lei Li. Learning kernel-smoothed machine translation with retrieved examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7280–7290, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.579. URL <https://aclanthology.org/2021.emnlp-main.579>. pages 214

Ting Jiang, Jian Jiao, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Denvy Deng, and Qi Zhang. Promptbert: Improving bert sentence embeddings with prompts. *ArXiv preprint*, abs/2201.04337, 2022b. URL <https://arxiv.org/abs/2201.04337>. pages 202

Zhipeng Jiang, Chao Zhao, Bin He, Yi Guan, and Jingchi Jiang. De-identification of medical records using conditional random fields and long short-term memory networks. *Journal of biomedical informatics*, 75: S43–S53, 2017. pages 3, 43

Xuyang Jin, Tao Ge, and Furu Wei. Plug and play knowledge distillation for kNN-LM with external logits. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 463–469, Online only, 2022. Association for Computational Linguistics. URL <https://aclanthology.org/2022.aacl-short.57>. pages 214

Alistair EW Johnson, Lucas Bulgarelli, and Tom J Pollard. Deidentification of free-text medical records using pre-trained bidirectional transformers. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, pages 214–221, 2020. pages 44, 45

- Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019. URL <https://ieeexplore.ieee.org/abstract/document/8733051/>. pages 219
- Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020a. doi: 10.1162/tacl\_a\_00300. URL <https://aclanthology.org/2020.tacl-1.5>. pages 84
- Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020b. doi: 10.1162/tacl\_a\_00300. URL <https://aclanthology.org/2020.tacl-1.5>. pages xiii, 85, 86, 97, 252
- Siddharth Karamcheti, Ranjay Krishna, Li Fei-Fei, and Christopher Manning. Mind your outliers! investigating the negative impact of outliers on active learning for visual question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7265–7281, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.564. URL <https://aclanthology.org/2021.acl-long.564>. pages 79
- Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL <https://openreview.net/forum?id=HklBjCEKvH>. pages 8, 199, 201, 207, 212, 213, 214
- Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Nearest neighbor machine translation. In *International Conference*

- on Learning Representations*, 2021. URL <https://openreview.net/forum?id=7wCB0fJ8hJM>. pages 214
- Imane Khaouja, Ismail Kassou, and Mounir Ghogho. A survey on skill identification from online job ads. *IEEE Access*, 9:118134–118153, 2021. pages 87, 89
- Kaung Khin, Philipp Burckhardt, and Rema Padman. A deep learning architecture for de-identification of patient notes: Implementation and evaluation. *ArXiv preprint*, abs/1810.01570, 2018. URL <https://arxiv.org/abs/1810.01570>. pages 44, 45
- Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1181. URL <https://aclanthology.org/D14-1181>. pages 176
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL <http://arxiv.org/abs/1412.6980>. pages 97
- Richard Kittredge and Ralph Grisham. *Analyzing Language in Restricted Domains: Sublanguage Description and Processing*. Lawrence Erlbaum Associates, 1986. pages 35
- Ilkka Kivimäki, Alexander Panchenko, Adrien Dessy, Dries Verdegem, Pascal Francq, Hugues Bersini, and Marco Saerens. A graph-based approach to skill extraction from text. In *Proceedings of TextGraphs-8 Graph-based Methods for Natural Language Processing*, pages 79–87, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL <https://aclanthology.org/W13-5011>. pages 87, 166, 213

- Alina Köchling and Marius Claus Wehner. Discriminated by an algorithm: a systematic review of discrimination and fairness by algorithmic decision-making in the context of hr recruitment and hr development. *Business Research*, 13(3):795–848, 2020. pages 185, 216
- Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand, 2005. URL <https://aclanthology.org/2005.mtsummit-papers.11>. pages 48
- AC Koivunen and AB Kostinski. The feasibility of data whitening to improve performance of weather radar. *Journal of Applied Meteorology and Climatology*, 38(6):741–749, 1999. URL <https://digitalcommons.mtu.edu/cgi/viewcontent.cgi?article=1279&context=physics-fp>. pages 202
- Dan Kondratyuk and Milan Straka. 75 languages, 1 model: Parsing Universal Dependencies universally. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2779–2795, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1279. URL <https://aclanthology.org/D19-1279>. pages 155
- Ioannis Konstantinidis, Manolis Maragoudakis, Ioannis Magnisalis, Christos Berberidis, and Vassilios Peristeras. Knowledge-driven unsupervised skills extraction for graph-based talent matching. *ArXiv preprint*, abs/10.1145, 2010. URL <https://arxiv.org/abs/10.1145>. pages 213
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. pages 1
- John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling



- sequence data. In Carla E. Brodley and Andrea Pohoreckyj Danyluk, editors, *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001)*, Williams College, Williamstown, MA, USA, June 28 - July 1, 2001, pages 282–289. Morgan Kaufmann, 2001. pages 26, 44, 98, 176
- J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977. pages 51, 94, 130
- Heiner Lasi, Peter Fettke, Hans-Georg Kemper, Thomas Feld, and Michael Hoffmann. Industry 4.0. *Business & information systems engineering*, 6: 239–242, 2014. pages 1
- Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.363. URL <https://aclanthology.org/2020.emnlp-main.363>. pages 169
- Martin le Vrang, Agis Papantoniou, Erika Pauwels, Pieter Fannes, Dominique Vandenstein, and Johan De Smedt. Esco: Boosting job matching in europe with semantic interoperability. *Computer*, 47(10):57–64, 2014. pages 5, 9, 12, 20, 85, 89, 124, 126, 152, 154, 167, 169, 232, 233, 252
- Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. pages 1
- Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020a. pages 45, 85, 102, 206

- Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 2020b. pages 172
- Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1612. URL <https://aclanthology.org/P19-1612>. pages 8, 213
- Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union, 1966. pages 130
- David D Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Machine learning proceedings 1994*, pages 148–156. Elsevier, 1994. pages 3, 4, 31, 62, 64, 251
- David D Lewis and William A Gale. A sequential algorithm for training text classifiers. In *SIGIR'94*, pages 3–12. Springer, 1994. pages 4, 62, 64, 251
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL <https://aclanthology.org/2020.acl-main.703>. pages 236, 237
- Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented

- generation for knowledge-intensive NLP tasks. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020b. URL <https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html>. pages 8, 213
- Belinda Z. Li, Sewon Min, Srinivasan Iyer, Yashar Mehdad, and Wentaoh Yih. Efficient one-pass end-to-end entity linking for questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6433–6441, Online, November 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.522. URL <https://aclanthology.org/2020.emnlp-main.522>. pages 240
- Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online, November 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.733. URL <https://aclanthology.org/2020.emnlp-main.733>. pages 202, 218
- Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online, 2020c. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.733. URL <https://aclanthology.org/2020.emnlp-main.733>. pages 202
- Shan Li, Baoxu Shi, Jaewon Yang, Ji Yan, Shuai Wang, Fei Chen, and Qi He. Deep job understanding at linkedin. In Jimmy Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu, editors, *Proceedings of the 43rd International ACM SIGIR conference on*

*research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 2145–2148. ACM, 2020d. doi: 10.1145/3397271.3401403. URL <https://doi.org/10.1145/3397271.3401403>. pages 87

Xin Li and Dan Roth. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002. URL <https://aclanthology.org/C02-1150>. pages 68, 70

Chen Ling, Xujiang Zhao, Jiaying Lu, Chengyuan Deng, Can Zheng, Junxiang Wang, Tanmoy Chowdhury, Yun Li, Hejie Cui, Xuchao Zhang, Tianjiao Zhao, Amit Panalkar, Wei Cheng, Haoyu Wang, Yanchi Liu, Zhengzhang Chen, Haifeng Chen, Chris White, Quanquan Gu, Jian Pei, and Liang Zhao. Domain specialization as the key to make large language models disruptive: A comprehensive survey. *ArXiv*, 2023. pages 35

Robert Litschko, Ivan Vulić, Simone Paolo Ponzetto, and Goran Glavaš. On cross-lingual retrieval with multilingual text encoders. *Information Retrieval Journal*, pages 1–35, 2022. pages 156, 157

Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. Deep learning for extreme multi-label text classification. In Noriko Kando, Tetsuya Sakai, Hideo Joho, Hang Li, Arjen P. de Vries, and Ryen W. White, editors, *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pages 115–124. ACM, 2017a. doi: 10.1145/3077136.3080834. URL <https://doi.org/10.1145/3077136.3080834>. pages 176

Liting Liu, Wenzheng Zhang, Jie Liu, Wenxuan Shi, and Yalou Huang. Learning multi-graph neural network for data-driven job skill prediction. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021. pages 87

- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, and Levy. Roberta: A robustly optimized bert pretraining approach. *ArXiv preprint*, abs/1907.11692, 2019. URL <https://arxiv.org/abs/1907.11692>. pages 153, 157, 169, 204, 205, 222
- Zengjian Liu, Yangxin Chen, Buzhou Tang, Xiaolong Wang, Qingcai Chen, Haodi Li, Jingfeng Wang, Qiwen Deng, and Suisong Zhu. Automatic de-identification of electronic medical records using token-level and character-level conditional random fields. *Journal of biomedical informatics*, 58:S47–S52, 2015. pages 3, 43
- Zengjian Liu, Buzhou Tang, Xiaolong Wang, and Qingcai Chen. De-identification of clinical notes via recurrent neural network and conditional random field. *Journal of biomedical informatics*, 75:S34–S42, 2017b. pages 44, 45
- Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. Zero-shot entity linking by reading entity descriptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3449–3460, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1335. URL <https://aclanthology.org/P19-1335>. pages 9, 29, 233, 234, 237
- Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL <https://openreview.net/forum?id=Bkg6RiCqY7>. pages 73, 97, 174, 219
- David Lowell, Zachary C. Lipton, and Byron C. Wallace. Practical obstacles to deploying active learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 21–30, Hong Kong, China, November 2019. Association

for Computational Linguistics. doi: 10.18653/v1/D19-1003. URL <https://aclanthology.org/D19-1003>. pages 63

Antoine Magron, Anna Dai, Mike Zhang, Syrielle Montariol, and Antoine Bosselut. JobSkape: A framework for generating synthetic job postings to enhance skill matching. In *Proceedings of the First Workshop on Natural Language Processing for Human Resources (NLP4HR 2024)*, pages 43–58, St. Julian’s, Malta, March 2024. Association for Computational Linguistics. URL <https://aclanthology.org/2024.nlp4hr-1.4>. pages

Emmanuel Malherbe and Marie-Aude Aufaure. Bridge the terminology gap between recruiters and candidates: A multilingual skills base built from social media and linked data. In Ravi Kumar, James Caverlee, and Hanghang Tong, editors, *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2016, San Francisco, CA, USA, August 18-21, 2016*, pages 583–590. IEEE Computer Society, 2016. doi: 10.1109/ASONAM.2016.7752295. URL <https://doi.org/10.1109/ASONAM.2016.7752295>. pages 139

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. Active learning by acquiring contrastive examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 650–663, 2021. pages 63, 66

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online, July 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.645. URL <https://aclanthology.org/2020.acl-main.645>. pages 52

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. CamemBERT: a tasty French language model. In *Proceedings of the 58th*

- Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online, 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.645. URL <https://aclanthology.org/2020.acl-main.645>. pages 177
- Héctor Martínez Alonso and Barbara Plank. When is multitask learning effective? semantic sequence prediction under varying data conditions. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 44–53, Valencia, Spain, 2017. Association for Computational Linguistics. URL <https://aclanthology.org/E17-1005>. pages 56
- Pedro Martins, Zita Marinho, and Andre Martins. Efficient machine translation domain adaptation. In *Proceedings of the 1st Workshop on Semiparametric Methods in NLP: Decoupling Logic from Knowledge*, pages 23–29, Dublin, Ireland and Online, 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.spanlp-1.3. URL <https://aclanthology.org/2022.spanlp-1.3>. pages 214
- Pedro Henrique Martins, Zita Marinho, and André FT Martins. Chunk-based nearest neighbor machine translation. *ArXiv preprint*, abs/2205.12230, 2022b. URL <https://arxiv.org/abs/2205.12230>. pages 214
- Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*, 2018. pages 255
- Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861, 2018. URL <https://www.theoj.org/joss-papers/joss.00861/10.21105.joss.00861.pdf>. pages 208
- Andrew J McMurry, Britt Fitch, Guergana Savova, Isaac S Kohane, and Ben Y Reis. Improved de-identification of physician notes through integrative modeling of both public and private medical text. *BMC medical informatics and decision making*, 13(1):112, 2013. pages 45

- Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947. URL [https://d1wqtxts1xzle7.cloudfront.net/48117997/bf0229599620160817-8553-1ubbzhr-libre.pdf?1471439562=&response-content-disposition=inline%3B+filename%3DNote\\_on\\_the\\_sampling\\_error\\_of\\_the\\_differ.pdf&Expires=1687321525&Signature=MndzUn74Uan2bisp02Fa2k~ycfrsKTpdonoDYT4oBZ~MamE6mB6qk9G341M-pb6NEjNh20awMt3CCoXVCKd3qHNYJNGvq5Hd6XtyaZdVDv5q4x6eXaPLvzZ~IB6TfiQRiCtriDzlk0uEw0Elo4CkWPQEBnEG7zpawqVvHG0Uo64Q1GCXGiqDDA8i71izRX7M6tImquqUcpGUPSPvBpaZfB1~ChQgEAHo7TIjH9EOEgW2WXtXWhwSItYvua3fDElrr2FdkHAKrRLZBscY00PbIZXYS3XDCyuc7hIQPUoB9H64Qe0bVlMguraV88ISxxLoPjLvzHOM8uW5nSgh0cTGQ\\_\\_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA](https://d1wqtxts1xzle7.cloudfront.net/48117997/bf0229599620160817-8553-1ubbzhr-libre.pdf?1471439562=&response-content-disposition=inline%3B+filename%3DNote_on_the_sampling_error_of_the_differ.pdf&Expires=1687321525&Signature=MndzUn74Uan2bisp02Fa2k~ycfrsKTpdonoDYT4oBZ~MamE6mB6qk9G341M-pb6NEjNh20awMt3CCoXVCKd3qHNYJNGvq5Hd6XtyaZdVDv5q4x6eXaPLvzZ~IB6TfiQRiCtriDzlk0uEw0Elo4CkWPQEBnEG7zpawqVvHG0Uo64Q1GCXGiqDDA8i71izRX7M6tImquqUcpGUPSPvBpaZfB1~ChQgEAHo7TIjH9EOEgW2WXtXWhwSItYvua3fDElrr2FdkHAKrRLZBscY00PbIZXYS3XDCyuc7hIQPUoB9H64Qe0bVlMguraV88ISxxLoPjLvzHOM8uW5nSgh0cTGQ__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA). pages 205
- Stephane M Meystre. De-identification of unstructured clinical data for patient privacy protection. In *Medical Data Privacy Handbook*, pages 697–716. Springer, 2015. pages 47
- Stephane M Meystre, F Jeffrey Friedlin, Brett R South, Shuying Shen, and Matthew H Samore. Automatic de-identification of textual documents in the electronic health record: a review of recent research. *BMC medical research methodology*, 10(1):70, 2010. pages 3, 43
- Sewon Min, Suchin Gururangan, Eric Wallace, Hannaneh Hajishirzi, Noah A Smith, and Luke Zettlemoyer. Silo language models: Isolating legal risk in a nonparametric datastore. *ArXiv preprint*, abs/2308.04430, 2023a. URL <https://arxiv.org/abs/2308.04430>. pages 214
- Sewon Min, Weijia Shi, Mike Lewis, Xilun Chen, Wen-tau Yih, Hannaneh Hajishirzi, and Luke Zettlemoyer. Nonparametric masked language modeling. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2097–2118, Toronto, Canada, July 2023b. Association for Computational Linguistics. URL <https://aclanthology.org/2023.findings-acl.132>. pages 214



- Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore, 2009. Association for Computational Linguistics. URL <https://aclanthology.org/P09-1113>. pages 6, 32, 125
- Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*, 2022. pages 255
- Dena F Mujtaba and Nihar R Mahapatra. Ethical considerations in ai-based recruitment. In *2019 IEEE International Symposium on Technology and Society (ISTAS)*, pages 1–7. IEEE, 2019. pages 185, 216
- Hiroki Nakayama. seqeval: A python framework for sequence labeling evaluation, 2018. URL <https://github.com/chakki-works/seqeval>. Software available from <https://github.com/chakki-works/seqeval>. pages 26
- Hiroki Nakayama, Takahiro Kubo, Junya Kamura, Yasufumi Taniguchi, and Xu Liang. doccano: Text annotation tool for human, 2018. URL <https://github.com/doccano/doccano>. Software available from <https://github.com/doccano/doccano>. pages 50, 94
- Ishna Neamatullah, Margaret M Douglass, H Lehman Li-wei, Andrew Reisner, Mauricio Villarroel, William J Long, Peter Szolovits, George B Moody, Roger G Mark, and Gari D Clifford. Automated de-identification of free-text medical records. *BMC medical informatics and decision making*, 8 (1):32, 2008. pages 44
- Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020*

- Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online, October 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.2. URL <https://aclanthology.org/2020.emnlp-demos.2>. pages 85, 102
- Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online, 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.2. URL <https://aclanthology.org/2020.emnlp-demos.2>. pages 172, 206
- Khanh Nguyen, Mike Zhang, Syrielle Montariol, and Antoine Bosselut. Rethinking skill extraction in the job market domain using large language models. In *Proceedings of the First Workshop on Natural Language Processing for Human Resources (NLP4HR 2024)*, pages 27–42, St. Julian's, Malta, March 2024. Association for Computational Linguistics. URL <https://aclanthology.org/2024.nlp4hr-1.3>. pages
- Joakim Nivre, Daniel Zeman, Filip Ginter, and Francis Tyers. Universal Dependencies. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts*, Valencia, Spain, 2017. Association for Computational Linguistics. URL <https://aclanthology.org/E17-5001>. pages 155
- Yasumasa Onoe and Greg Durrett. Fine-grained entity typing for domain independent entity linking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8576–8583, 2020. pages 9, 233
- OpenAI. Chatgpt (march version), 2023. URL <https://chat.openai.com/chat>. pages 1, 235
- Maria Papoutsoglou, Nikolaos Mittas, and Lefteris Angelis. Mining people analytics from stackoverflow job advertisements. In *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 108–115. IEEE, 2017. pages 139

- Namrata Patel, Pierre Accorsi, Diana Inkpen, Cédric Lopez, and Mathieu Roche. Approaches of anonymisation of an sms corpus. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 77–88. Springer, 2013. pages 3, 45, 46
- Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL <https://aclanthology.org/D14-1162>. pages 52
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL <https://aclanthology.org/N18-1202>. pages 45, 125
- Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1005. URL <https://aclanthology.org/D19-1005>. pages 7, 182
- Wilhelm H Peterßen. Kleines methoden-lexikon. 2. aktualisierte aufl, 2001. pages 20
- Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean

- Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.200. URL <https://aclanthology.org/2021.naacl-main.200>. pages 8, 213
- Barbara Plank, Anders Søgaard, and Yoav Goldberg. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 412–418, Berlin, Germany, 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-2067. URL <https://aclanthology.org/P16-2067>. pages 51, 52, 250
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018. pages 125
- Manish Raghavan, Solon Barocas, Jon Kleinberg, and Karen Levy. Mitigating bias in algorithmic hiring: Evaluating claims and practices. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 469–481, 2020. pages 185, 216
- Jonathan Raiman and Olivier Raiman. Deeptype: Multilingual entity linking by neural type system evolution. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5406–5413. AAAI Press, 2018. URL <https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17148>. pages 9, 233

- Ori Ram, Yuval Kirstain, Jonathan Berant, Amir Globerson, and Omer Levy. Few-shot question answering by pretraining span selection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3066–3079, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.239. URL <https://aclanthology.org/2021.acl-long.239>. pages 97
- Abhinav Ramesh Kashyap, Devamanyu Hazarika, Min-Yen Kan, and Roger Zimmermann. Domain divergences: A survey and empirical analysis. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1830–1849, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.147. URL <https://aclanthology.org/2021.naacl-main.147>. pages 35
- Alan Ramponi and Barbara Plank. Neural unsupervised domain adaptation in NLP—A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6838–6855, Barcelona, Spain (Online), 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.603. URL <https://aclanthology.org/2020.coling-main.603>. pages 35
- Lance Ramshaw and Mitch Marcus. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*, 1995. URL <https://aclanthology.org/W95-0107>. pages 25, 26
- Felix Rauner, Rupert Maclean, et al. *Handbook of technical and vocational education and training research*, volume 49. Springer, 2008. pages 19
- Nils Reimers and Iryna Gurevych. Why comparing single performance scores does not allow to draw conclusions about machine learnin. *ArXiv preprint*, abs/1803.09578, 2018. URL <https://arxiv.org/abs/1803.09578>. pages 74, 117, 143

- Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL <https://aclanthology.org/D19-1410>. pages 243
- Michael T Rosenstein, Zvika Marx, Leslie Pack Kaelbling, and Thomas G Dietterich. To transfer or not to transfer. In *NIPS 2005 workshop on transfer learning*, volume 898, pages 1–4, 2005. pages 136
- Sebastian Ruder. An overview of multi-task learning in deep neural networks. *ArXiv preprint*, abs/1706.05098, 2017. URL <https://arxiv.org/abs/1706.05098>. pages 35, 37, 57
- Javier Sánchez-Monedero, Lina Dencik, and Lilian Edwards. What does it mean to solve the problem of discrimination in hiring? social, technical and legal perspectives from the uk on automated hiring systems. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 458–468, 2020. pages 185, 216
- Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021. pages 255
- Danielle Saunders. Domain adaptation and multi-domain adaptation for neural machine translation: A survey. *Journal of Artificial Intelligence Research*, 75:351–424, 2022. pages 35
- Luiza Sayfullina, Eric Malmi, and Juho Kannala. Learning representations for soft skill matching. In *International Conference on Analysis of Images, Social Networks and Texts*, pages 141–152, 2018. pages xvi, 6, 84, 87, 88, 124, 138, 152, 154, 166, 176, 204, 213

- Klaus Schwab. *The fourth industrial revolution*. Currency, 2017. pages 1
- Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL <https://openreview.net/forum?id=H1aIuk-RW>. pages 4, 62, 66, 251
- Elena Senger, Mike Zhang, Rob van der Goot, and Barbara Plank. Deep learning-based computational job market analysis: A survey on skill extraction and classification from job postings. In *Proceedings of the First Workshop on Natural Language Processing for Human Resources (NLP4HR 2024)*, pages 1–15, St. Julian's, Malta, March 2024. Association for Computational Linguistics. URL <https://aclanthology.org/2024.nlp4hr-1.1>. pages
- Burr Settles. Active learning literature survey. 2009. pages 3, 31, 64
- Burr Settles. Active learning. *Synthesis lectures on artificial intelligence and machine learning*, 6(1):1–114, 2012. pages 64
- Claude E Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948. pages 64
- Yanyao Shen, Hyokun Yun, Zachary C. Lipton, Yakov Kronrod, and Animesh Anandkumar. Deep active learning for named entity recognition. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL <https://openreview.net/forum?id=ry018WZAZ>. pages 63
- Baoxu Shi, Jaewon Yang, Feng Guo, and Qi He. Saliency and market-aware skill extraction for job targeting. In Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash, editors, *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA*,

- USA, August 23-27, 2020, pages 2871–2879. ACM, 2020. URL <https://dl.acm.org/doi/10.1145/3394486.3403338>. pages 87, 166, 213
- Weijia Shi, Julian Michael, Suchin Gururangan, and Luke Zettlemoyer. Nearest neighbor zero-shot inference. *ArXiv preprint*, abs/2205.13792, 2022. URL <https://arxiv.org/abs/2205.13792>. pages 214
- Elisa Margareth Sibarani, Simon Scerri, Camilo Morales, Sören Auer, and Diego Collarana. Ontology-guided job market demand analysis: a cross-sectional study for the data science field. In *Proceedings of the 13th International Conference on Semantic Systems*, pages 25–32, 2017. pages 139
- Aditya Siddhant and Zachary C. Lipton. Deep Bayesian active learning for natural language processing: Results of a large-scale empirical study. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2904–2909, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1318. URL <https://aclanthology.org/D18-1318>. pages 64, 78
- Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. Large language models encode clinical knowledge. *Nature*, pages 1–9, 2023. pages 35
- Ellery Smith, Martin Braschler, Andreas Weiler, and Thomas Haberthuer. Syntax-based skill extractor for job advertisements. In *2019 6th Swiss Conference on Data Science (SDS)*, pages 80–81. IEEE, 2019. pages 87, 166, 213
- Ellery Smith, Andreas Weiler, and Martin Braschler. Skill extraction for domain-specific text retrieval in a job-matching platform. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 116–128. Springer, 2021. pages 87



Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. Portuguese named entity recognition using bert-crf. *ArXiv preprint*, abs/1909.10649, 2019. URL <https://arxiv.org/abs/1909.10649>. pages 55, 98

Georg Spöttl and Lars Windelband. The 4th industrial revolution—its impact on vocational skills. *Journal of Education and Work*, 34(1):29–52, 2021. pages 249

Andreas Stephan and Benjamin Roth. WeaNF””: weak supervision with normalizing flows. In *Proceedings of the 7th Workshop on Representation Learning for NLP*, pages 269–279, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.repl4nlp-1.27. URL <https://aclanthology.org/2022.repl4nlp-1.27>. pages 32

Amber Stubbs and Özlem Uzuner. Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/uthealth corpus. *Journal of biomedical informatics*, 58:S20–S29, 2015a. pages 177

Amber Stubbs and Özlem Uzuner. Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/UTHealth corpus. *Journal of Biomedical Informatics*, 58:S20–S29, 2015b. ISSN 1532-0464. doi: 10.1016/j.jbi.2015.07.020. URL <http://www.sciencedirect.com/science/article/pii/S1532046415001823>. pages 44, 49, 53, 250

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. Whitening sentence representations for better semantics and faster retrieval. *ArXiv preprint*, abs/2103.15316, 2021. URL <https://arxiv.org/abs/2103.15316>. pages 201, 202, 218

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hananeh Hajishirzi, Noah A. Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online, November 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.746.

URL <https://aclanthology.org/2020.emnlp-main.746>. pages 10, 62, 66, 67, 68, 75, 76, 77, 79, 80, 251

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hananeh Hajishirzi, Noah A. Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online, 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.746. URL <https://aclanthology.org/2020.emnlp-main.746>. pages 4

György Szarvas, Richárd Farkas, and Róbert Busa-Fekete. State-of-the-art anonymization of medical records using an iterative machine learning framework. *Journal of the American Medical Informatics Association*, 14 (5):574–580, 2007. pages 3, 42, 43

Jeniya Tabassum, Mounica Maddela, Wei Xu, and Alan Ritter. Code and named entity recognition in StackOverflow. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4913–4926, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.443. URL <https://aclanthology.org/2020.acl-main.443>. pages 43, 52

Damian A Tamburri, Willem-Jan Van Den Heuvel, and Martin Garriga. Dataops for societal intelligence: a data pipeline for labor market skills extraction and matching. In *2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI)*, pages 391–394. IEEE, 2020. pages 6, 87, 88, 124, 138, 152, 166, 213

Erik F. Tjong Kim Sang. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*, 2002a. URL <https://aclanthology.org/W02-2024>. pages 53

Erik F. Tjong Kim Sang. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The*

- 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*, 2002b. URL <https://aclanthology.org/W02-2024>. pages 26, 98, 250
- Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003. URL <https://aclanthology.org/W03-0419>. pages 26
- Jean-Francois Ton, Walter Talbott, Shuangfei Zhai, and Joshua M. Susskind. Regularized training of nearest neighbor language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop*, pages 25–30, Hybrid: Seattle, Washington + Online, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-srw.4. URL <https://aclanthology.org/2022.naacl-srw.4>. pages 214
- Maaske Treurniet, Orphée De Clercq, Henk van den Heuvel, and Nelleke Oostdijk. Collection of a corpus of Dutch SMS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2268–2273, Istanbul, Turkey, 2012. European Language Resources Association (ELRA). URL [http://www.lrec-conf.org/proceedings/lrec2012/pdf/537\\_Paper.pdf](http://www.lrec-conf.org/proceedings/lrec2012/pdf/537_Paper.pdf). pages 3, 45, 46
- J Trienes, D Trieschnigg, C Seifert, and D Hiemstra. Comparing rule-based, feature-based and deep neural methods for de-identification of dutch medical records. In *Eickhoff, C.(ed.), Health Search and Data Mining Workshop: Proceedings of the ACM WSDM 2020 Health Search and Data Mining Workshop co-located with the 13th ACM International WSDM Conference (WSDM 2020) Houston, Texas, USA, February 3, 2020*, pages 3–11. [SI]: CEUR, 2020. pages 3, 42, 43, 44, 45, 51
- Dennis Ulmer. deep-significance: Easy and Better Significance Testing for Deep Neural Networks, 2021. URL <https://doi.org/10.5281/zeno>

do.4638709. <https://github.com/Kaleidophon/deep-significance>. pages 74, 117, 143

Dennis Ulmer, Elisa Bassignana, Max Müller-Eberstein, Daniel Varab, Mike Zhang, Rob van der Goot, Christian Hardmeier, and Barbara Plank. Experimental standards for deep learning in natural language processing research. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2673–2692, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL <https://aclanthology.org/2022.findings-emnlp.196>. pages

Rob van der Goot, Ibrahim Sharaf, Aizhan Imankulova, Ahmet Üstün, Marija Stepanović, Alan Ramponi, Siti Oryza Khairunnisa, Mamoru Komachi, and Barbara Plank. From masked language modeling to translation: Non-English auxiliary tasks improve zero-shot spoken language understanding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2479–2497, Online, 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.197. URL <https://aclanthology.org/2021.naacl-main.197>. pages 159, 164

Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. Massive choice, ample tasks (MaChAmp): A toolkit for multi-task learning in NLP. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 176–197, Online, April 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-demos.22. URL <https://aclanthology.org/2021.eacl-demos.22>. pages 99, 115, 134, 142, 155, 157, 195

Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. Massive choice, ample tasks (MaChAmp): A toolkit for multi-task learning in NLP. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics:*

- System Demonstrations*, pages 176–197, Online, 2021c. Association for Computational Linguistics. URL <https://aclanthology.org/2021.eacl-demos.22>. pages 52
- Sarah-Jane van Els, David Graus, and Emma Beauxis-Aussalet. Improving fairness assessments with synthetic data: a practical use case with a recommender system for human resources. 2022. pages 185, 216
- C Van Rijsbergen. Information retrieval: theory and practice. In *Proceedings of the Joint IBM/University of Newcastle upon Tyne Seminar on Data Base Systems*, volume 79, 1979. pages 27
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL <https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html>. pages 177
- Dexin Wang, Kai Fan, Boxing Chen, and Deyi Xiong. Efficient cluster-based  $k$ -nearest-neighbor machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2175–2187, Dublin, Ireland, 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.154. URL <https://aclanthology.org/2022.acl-long.154>. pages 214
- Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1405–1418, Online, 2021a. Association for Computational Linguistics. doi:

10.18653/v1/2021.findings-acl.121. URL <https://aclanthology.org/2021.findings-acl.121>. pages 7, 182

Shuhe Wang, Xiaoya Li, Yuxian Meng, Tianwei Zhang, Rongbin Ouyang, Jiwei Li, and Guoyin Wang. KNN-NER: Named entity recognition with nearest neighbor search. *ArXiv preprint*, abs/2203.17103, 2022b. URL <https://arxiv.org/abs/2203.17103>. pages 201, 214

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194, 2021b. doi: 10.1162/tacl\_a\_00360. URL <https://aclanthology.org/2021.tacl-1.11>. pages 7, 182

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=gEzrGCozdqR>. pages 255

Christo Wilson, Avijit Ghosh, Shan Jiang, Alan Mislove, Lewis Baker, Janelle Szary, Kelly Trindel, and Frida Polli. Building and auditing fair algorithms: A case study in candidate screening. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 666–677, 2021. pages 185, 216

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020a. Association for

Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL <https://aclanthology.org/2020.emnlp-demos.6>. pages 86, 97, 155, 172

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL <https://aclanthology.org/2020.emnlp-demos.6>. pages 219

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.519. URL <https://aclanthology.org/2020.emnlp-main.519>. pages 9, 29, 232, 233, 234, 235, 237

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. Bloomberggpt: A large language model for finance. *ArXiv*, 2023. pages 35

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, and Macherey. Google’s neural machine translation system: Bridging the gap between human and machine translation. *ArXiv preprint*, abs/1609.08144, 2016. URL <https://arxiv.org/abs/1609.08144>. pages 97

Frank F Xu, Uri Alon, and Graham Neubig. Why do nearest neighbor

- language models work? *ArXiv preprint*, abs/2301.02828, 2023. URL <https://arxiv.org/abs/2301.02828>. pages 214
- Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. ConSERT: A contrastive framework for self-supervised sentence representation transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5065–5075, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.393. URL <https://aclanthology.org/2021.acl-long.393>. pages 202
- Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California, 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1174. URL <https://aclanthology.org/N16-1174>. pages 176
- Michihiro Yasunaga, Jure Leskovec, and Percy Liang. LinkBERT: Pretraining language models with document links. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8003–8016, Dublin, Ireland, May 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.551. URL <https://aclanthology.org/2022.acl-long.551>. pages 172, 174, 182
- Michihiro Yasunaga, Jure Leskovec, and Percy Liang. Linkbert: Pretraining language models with document links. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8003–8016, 2022b. pages 253
- Wenbiao Yin and Lin Shang. Efficient nearest neighbor emotion classification with BERT-whitening. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4738–4745,



- Abu Dhabi, United Arab Emirates, 2022. Association for Computational Linguistics. URL <https://aclanthology.org/2022.emnlp-main.312>. pages 201, 202, 214
- Dani Yogatama, Cyprien de Masson d’Autume, and Lingpeng Kong. Adaptive semiparametric language models. *Transactions of the Association for Computational Linguistics*, 9:362–373, 2021. doi: 10.1162/tacl\_a\_00371. URL <https://aclanthology.org/2021.tacl-1.22>. pages 214
- Wenhao Yu, Chenguang Zhu, Yuwei Fang, Donghan Yu, Shuohang Wang, Yichong Xu, Michael Zeng, and Meng Jiang. Dict-BERT: Enhancing language model pre-training with dictionary. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1907–1918, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.150. URL <https://aclanthology.org/2022.findings-acl.150>. pages 7, 182
- Klim Zaporozhets, Lucie-Aimée Kaffee, Johannes Deleu, Thomas Demeester, Chris Develder, and Isabelle Augenstein. Tempel: Linking dynamically evolving and newly emerging entities. *Advances in Neural Information Processing Systems*, 35:1850–1866, 2022. pages 29, 237
- Mike Zhang and Barbara Plank. Cartography active learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 395–406, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.36. URL <https://aclanthology.org/2021.findings-emnlp.36>. pages
- Mike Zhang, Kristian Jensen, Sif Sonniks, and Barbara Plank. SkillSpan: Hard and soft skill extraction from English job postings. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4962–4984, Seattle, United States, July 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.366. URL <https://aclanthology.org/2022.naacl-main.366>. pages xvi, 128, 130, 133,

152, 153, 154, 158, 166, 175, 203, 204, 205, 206, 213, 219, 220, 222, 235

Mike Zhang, Kristian Nørgaard Jensen, and Barbara Plank. Kompetencer: Fine-grained skill classification in Danish job postings via distant supervision and transfer learning. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 436–447, Marseille, France, June 2022b. European Language Resources Association. URL <https://aclanthology.org/2022.lrec-1.46>. pages 9, 233

Mike Zhang, Kristian Nørgaard Jensen, and Barbara Plank. Kompetencer: Fine-grained skill classification in danish job postings via distant supervision and transfer learning. In *Proceedings of the Language Resources and Evaluation Conference*, pages 436–447, Marseille, France, 2022c. European Language Resources Association. URL <https://aclanthology.org/2022.lrec-1.46>. pages xvi, 152, 166, 167, 175, 213

Mike Zhang, Kristian Nørgaard Jensen, Rob van der Goot, and Barbara Plank. Skill extraction from job postings using weak supervision. In *Proceedings of RecSysHR'22*. RecSysHR'22, 2022d. URL [https://ceur-ws.org/Vol-3218/RecSysHR2022-paper\\_10.pdf](https://ceur-ws.org/Vol-3218/RecSysHR2022-paper_10.pdf). pages 166, 179, 213

Mike Zhang, Rob van der Goot, and Barbara Plank. ESCOXML-R: Multilingual Taxonomy-driven Pre-training for the Job Market Domain. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11871–11890, Toronto, Canada, 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.662. URL <https://aclanthology.org/2023.acl-long.662>. pages

Mike Zhang, Rob van der Goot, and Barbara Plank. ESCOXML-R: Multilingual taxonomy-driven pre-training for the job market domain. *ArXiv preprint*, abs/2305.12092, 2023b. URL <https://arxiv.org/abs/2305.12092>. pages 213

- Mike Zhang, Rob van der Goot, Min-Yen Kan, and Barbara Plank. NNOSE: Nearest neighbor occupational skill extraction. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 589–608, St. Julian’s, Malta, March 2024a. Association for Computational Linguistics. URL <https://aclanthology.org/2024.eacl-long.35>. pages
- Mike Zhang, Rob van der Goot, and Barbara Plank. Entity linking in the job market domain. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 410–419, St. Julian’s, Malta, March 2024b. Association for Computational Linguistics. URL <https://aclanthology.org/2024.findings-eacl.28>. pages
- Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657, 2015. URL <https://proceedings.neurips.cc/paper/2015/hash/250cf8b51c773f3f8dc8b4be867a9a02-Abstract.html>. pages 68, 70
- Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1139. URL <https://aclanthology.org/P19-1139>. pages 7, 182
- Meng Zhao, Faizan Javed, Ferosh Jacob, and Matt McNair. SKILL: A system for skill identification and normalization. In Blai Bonet and Sven Koenig, editors, *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 4012–4018. AAAI Press, 2015. URL <http://www.aaai.org/ocs/index.php/IAAI/IAAI15/paper/view/9363>. pages 87, 166, 213

- Zhiqun Zhao and Felix Rauner. *Areas of vocational education research*. Springer, 2014. pages 19
- Fedor Zhdanov. Diverse mini-batch active learning. *ArXiv preprint*, abs/1901.05954, 2019. URL <https://arxiv.org/abs/1901.05954>. pages 4, 62, 78, 251
- Xin Zheng, Zhirui Zhang, Junliang Guo, Shujian Huang, Boxing Chen, Weihua Luo, and Jiajun Chen. Adaptive nearest neighbor machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 368–374, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-short.47. URL <https://aclanthology.org/2021.acl-short.47>. pages 214
- Dawei Zhu, Xiaoyu Shen, Marius Mosbach, Andreas Stephan, and Dietrich Klakow. Weaker than you think: A critical look at weakly supervised learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14229–14253, Toronto, Canada, July 2023a. Association for Computational Linguistics. URL <https://aclanthology.org/2023.acl-long.796>. pages 32
- Fangwei Zhu, Jifan Yu, Hailong Jin, Lei Hou, Juanzi Li, and Zhifang Sui. Learn to not link: Exploring NIL prediction in entity linking. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10846–10860, Toronto, Canada, July 2023b. Association for Computational Linguistics. URL <https://aclanthology.org/2023.findings-acl.690>. pages 244
- Wenhao Zhu, Shujian Huang, Yunzhe Lv, Xin Zheng, and Jiajun Chen. What knowledge is needed? towards explainable memory for knnmt domain adaptation. *ArXiv preprint*, abs/2211.04052, 2022. URL <https://arxiv.org/abs/2211.04052>. pages 214

Wenhao Zhu, Qianfeng Zhao, Yunzhe Lv, Shujian Huang, Siheng Zhao, Sizhe Liu, and Jiajun Chen. knn-box: A unified framework for nearest neighbor generation. *ArXiv preprint*, abs/2302.13574, 2023c. URL <https://arxiv.org/abs/2302.13574>. pages 214

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 19–27. IEEE Computer Society, 2015. doi: 10.1109/ICCV.2015.11. URL <https://doi.org/10.1109/ICCV.2015.11>. pages 97