Better Language Support and Pragmatic System Designs for Automatic Text Summarization

Daniel Varab

A thesis presented for the degree of
Doctor of Philosophy

Supervisors: Christian Hardmeier, Anders Markussen

Department of Computer Science
IT University of Copenhagen

Quality Data & Insights
Novo Nordisk

Denmark
March, 2023

IT UNIVERSITY OF COPENHAGEN

Abstract

Automatic text summarization is a promising technology that has gained noticeable traction in recent years as a result of innovations in computational hardware and increased accessibility to large pre-trained language models. Despite apparent progress, state-of-the-art summarization research remains an unbalanced technology with an overwhelming Anglo-centric focus and a tendency to prioritize complexity over simplicity when designing systems. For summarization to ease the information challenges faced by society today, research must move beyond the scope of a single language and develop systems that target a wider range of languages. We as researchers must ensure the availability of methods that are compatible with the languages that practitioners need to support and make a sincere effort to identify users' needs and develop pragmatic systems which serve them.

This thesis argues that current summarization research focuses on improving summarizers that are only applicable to a few settings, leaving fundamental questions about what can realistically be expected of summarization technology in a broader context unanswered. It argues that progress in systems design should not only strive to specialize but value general-purpose designs that meet the needs of practitioners. Centrally, for summarization to be accepted as a reliable language technology, we must make a serious effort to ensure that systems work reliably in many languages, settings, and domains to convincingly aid real-world challenges.

The thesis is formatted as an article-based thesis and represents the research findings of a three-year industrial Ph.D., resulting in five academic papers. The contribution of this work is the concrete development of three large-scale datasets and two case studies that emphasize the effectiveness of simple system designs. The findings pave the way for future research of multilingual summarization research and show that not only can more simple systems be competitive, but they can outperform more complex designs while remaining easy to implement, extend, and benefit from progress in adjacent NLP tasks.

Resumé

Automatisk tekstopsummering er en lovende teknologi, der gennem de seneste år har været udsat for en eksplosiv udvikling som følge af udvikling inden for hardware og øget tilgængelighed til store præ-trænede sprogmodeller. På trods af hvad der umiddelbart ligner fremskridt, er forskning inden tekstopsummering blevet en skæv teknologi med et overvældende fokus på Engelsk og med en tendens til at fortrække komplekse frem for enkle systemløsninger. Hvis opsummeringsteknologi skal hjælpe med samfundets informationsudfordringer skal forskning gøre op med at fokusere på blot et enkelt sprog og udvikle systemer der understøtter et bredere udvalg af sprog. Vi som forskere skal sikre at metoder er kompatible med de sprog som brugere har brug for at understøtte og gøre en oprigtig indsats for at udvikle pragmatiske systemer, der tjener brugernes behov.

Denne afhandling argumenterer, at nuværende forskning inden for tesktopsummering fokuserer for meget på forbedring modeler inden for én bestemt niche, hvilket efterlader grundlæggende spørgsmål omkring, hvad der i virkeligheden kan forventes af opsummeringsteknologi i en bredere sammenhæng. Den argumenterer, at fremskridt inden for tesktopsummering ikke kun bør stræbe efter at specialisere sig, men værdsætte alsidige modeler som løser flere af brugernes behov på samme tid. Hvis tekstopsummering skal accepteres som en pålidelig sprogteknologi, må vi gøre en seriøs indsats for at sikre, at systemer fungerer pålideligt på tværs af forkellige sprog, konfigurationer, og domæner for at løse rigtige udfordringer der findes i dagligdagen.

Denne afhandlingen er formateret som en artikelbaseret afhandling og indeholder resultaterne af en treårig erhvervs-ph.d., der resulterer i fem akademiske artikler. Afhandlingen bidrager konkret med tre store datasæt og to studier, der understreger fordelene ved simple system løsninger. Disse bidrag baner vejen for fremtidig forskning inden for flersproget tekstopsummering og viser, at simple systemer ud over at være nemme at implementere kan være konkurrencedygtige, såvel sagtens kan udkonkurrere komplekse systemer.

Acknowledgements

I would like to express my sincere gratitude to my academic thesis advisors, Natalie Schluter and Christian Hardmeier, for their support, guidance, and encouragement throughout my years of study. Their experience and advice have been invaluable in shaping my view on science and helped me develop my understanding of the role of natural language processing. I would also like to thank all of my industry supervisors for their support throughout this industrial Ph.D. project, and express a particular thanks to Kenneth Petersen and Anders Markussen, for their persistent advice despite turmoil circumstances. For their valuable feedback and suggestions, I am grateful to my colleagues for their friendship, support, and daily (non) work-related discussions. I am also grateful to Novo Nordisk and Innovation Fund Denmark for providing financial support for this research. I would like to thank my family and friends for their love and support throughout these past years. Finally, a special thanks to you Ida. Without your reflections and support during the highs and lows, this accomplishment would not even have been remotely possible.

Industry Supervisors Christian Frimundt Petersen, Thomas Møller Holbek, Vivian Fussing, Kenneth Petersen, Kasper Bøwig Rasmussen, Morten Rune Nielsen, Anders Markussen

Academic Supervisors Associate Professor Christian Hardmeier, Associate Professor Natalie Schluter

Declaration of Work

I declare that I am the sole author of this thesis and that the content presented herein is entirely my original work. I confirm that this thesis has not been submitted for qualifications at any other academic institution in Denmark or abroad.

- *Daniel Varab*

Preface

This thesis represents the combined work conducted during a three-year industrial Ph.D. born out of a collaboration between Novo Nordisk, Innovation Fund Denmark, and the IT University of Copenhagen starting in January 2020. The work focuses on improving methods for automatic text summarization, a subfield of the natural language processing field. The content of this thesis is written with the assumption that the reader has some basic understanding of NLP, and standard supervised learning models. For readers well-versed in summarization literature, large parts of the thesis (Chapter 2) will likely be trivial at times, however, I hope it may present alternative views on the subject. Finally, thanks for taking the time to read this work, I hope you may find it both insightful as well as thought-provoking and that it may lie as the basis for interesting discussions and future research.

- *Daniel Varab*

Included Papers

1. Daniel Varab and Natalie Schluter. 2020. **DaNewsroom: A Large-scale Danish Summarisation Dataset**. *In Proceedings of the 12. Language Resources and Evaluation Conference*, pages 6731-6739, Marseille, France. European Language Resources Association.

2. Daniel Varab and Natalie Schluter. 2021. **MassiveSumm: a very large-scale, very multilingual, news summarisation dataset**. *In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10150-10161, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

3. Leon Strømberg-Derczynski, Manuel Ciosici, Rebekah Baglini, Morten H. Christiansen, Jacob Aarup Dalsgaard, Riccardo Fusaroli, Peter Juel Henrichsen, Rasmus Hvingelby, Andreas Kirkedal, Alex Speed Kjeldsen, Claus Ladefoged, Finn Årup Nielsen, Jens Madsen, Malte Lau Petersen, Jonathan Hvithamar Rystrøm, and Daniel Varab. 2021. **The Danish Gigaword Corpus**. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 413-421, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

4. Daniel Varab and Yumo Xu. 2023. **Abstractive Summarizers are Excellent Extractive Summarizers**. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 330–339, Toronto, Canada. Association for Computational Linguistics.

5. Daniel Varab and Christian Hardmeier. **With Good MT There is No Need For End-to-End: A Case for Translate-then-Summarize Cross-lingual Summarization**. *Conference paper under review.*

Contents

Chapter 1

Introduction

1.1 Project Motivation

Today's society is flooded with information. Whether it is information stored in news articles, legal documents, or business reports, it is increasingly difficult to digest information at the pace at which it is being produced. This has two consequences, either information is too hard to find or it is entirely overlooked. As a result, there is a growing demand for automated tools that can efficiently alleviate the information overload people, companies, and society all together are experiencing today. By reducing the amount of text to read, automatic text summarization provides a promising solution that allows people to quickly identify relevant information in large collections of documents. Furthermore, by producing summaries of text documents, readers generally save time and effort without compromising key information hidden away in documents.

Summarizers have improved significantly in recent years with the increased ease of generating text and performance through sequence-to-sequence models and accessible pre-trained language models. However, recent success stories of summarization technology must be considered with some degree of caution. Notably, improvements to state-of-the-art summarization research apply primarily to English summarizers and rely on complex neural networks that require large amounts of data that do not exist for many languages.

The reality of text summarization technology is that it is still immature and that few languages benefit from ongoing research. Even for the few languages, designs remain turbulent and are dominated by custom architectures that are largely incompatible with one another, preventing incremental improvements.

1.2 Contributions

Summarization technology is increasingly reliant on data-driven methods and algorithms. With most summarization research being based on English experiments, the reality is that few languages have access to competitive summarizers or datasets. This makes it increasingly unclear to which degree innovation in summarization generalizes beyond applications to English. This thesis hypothesizes that the lack of language diversity can be remedied by facilitating summarization datasets for a wider range of languages. Currently, several automatic methods to create summarization datasets exist for English, however, it is unsure whether these methods can successfully be applied to other languages. This thesis explores this research opportunity and contributes with the following:

1. A one billion-word corpus of Danish text, containing high-quality text data sampled from a wide range of representative text domains. The dataset enables future investigations into data-intensive methods (for the Danish language) which have become a necessity for state-of-the-art language technology tools.

2. The first automatic summarization dataset for the Danish language with more than 1.1 million news articles paired with manually written summaries. The large-scale dataset enables future research on building Danish summarizers that rely on data-intensive techniques.

3. A language-agnostic data collection method for summarization, showing that existing English methods can be modified to support other languages with relative ease and successfully produce large datasets *for some languages*.

4. The largest and most diverse multilingual summarization dataset to date, covering 92 languages, 38 language families and 35 scripts. This massive summarization dataset facilitates languages to a wide range of languages that previously had no data, and improves the conditions of future multilingual summarization.

Cutting-edge summarization systems have presented impressive results in recent years but pose practical challenges with the side effects of a fragmented ecosystem. With each newly proposed system practitioners are confronted with new architectures that are incompatible with prior established systems. This is a result of a research trend that tends to favor design complexity

over simplicity. This thesis argues that complexity is a design choice, not a necessity, and hypothesizes that simple systems can be just as competitive as more complex systems. To explore this hypothesis and counter an ongoing research trend this thesis contributes with the following:

1. A unified summarization system that supports producing both extractive *and* abstractive summaries. The system achieves this without compromising performance, exhibiting competitive performance across both summary types compared to other specialized systems. This is achieved through a simple but powerful inference technique and represents the first model of its kind and stands as a paradigm shift in summarization system designs, paving the way for future multi-summary-type systems.

2. A comparative review between end-to-end and pipeline systems for cross-lingual summarization. The study reviews the paradigm's efficacy and demonstrates that simple pipeline systems exhibit strong performance, capitalizing on individual progress in machine translation and monolingual summarization. It concludes that contrary to successes seen in other NLP tasks, end-to-end text generation systems do not convincingly outperform pipeline methods.

1.3 Thesis Outline

This thesis is organized as follows:

- **Chapter 2** is a background section that provides a brief introduction to text summarization to give the necessary context to discuss the contributions included in this thesis. It includes a definition of a summary, the task, evaluation, necessary tools, data assumptions and a brief enumeration of contemporary systems.
- **Chapter 3** introduces, motivates, and outlines the contributions made to multilingual data resource creation. It presents three papers that describe the creation of large-scale datasets; The first Danish text corpora, the first Danish summarization dataset, and the largest and most diverse multilingual summarization dataset to date.
- **Chapter 4** motivates and outlines the contributions made to pragmatic summarization system designs. It includes two studies; One presenting the first unified summarization system that can produce both extractive and abstractive summaries, and a second study arguing against the necessity of end-to-end designs in cross-lingual summarization.
- **Chapter 5** briefly discusses strengths, limitations, and biases of the contributions in this thesis and presents insights obtained during the work on the thesis. This is followed by suggestions for future potential research directions.

Chapter 2

A Summary of Text Summarization

2.1 Defining Summarization

A text summary is a brief and concise version of a longer document. It conveys the most important information, and key points from the source document and omits redundant information and points irrelevant to the user. A summary's purpose is to give the reader an overview of the main points of the original material, without having to read the entire thing, thus, reducing the time to retrieve information, increasing efficiency and lowering costs. The motivation for the text summarization was initially framed by Luhn (1958):

> *The objective is to automate the process of producing an abstract to save a prospective reader time and effort in finding useful information in a given article or report.*

This has in more recent years been broadened to the goal of finding information and conveying the information according to users needs by Mani and Maybury (2002):

> *The goal of automatic summarization is to take an information source, extract content from it, and present the most important content in a condensed form and in a manner sensitive to the user's or application's needs.*

A summary, therefore, serves to save a user's time and can do so in one of two ways as either an *indicative summary* or an *informative summary* (Edmundson, 1969; Borko and Bernier, 1975). The goal of an indicative summary is to aid an information searcher in browsing a collection of documents to decide which documents are relevant. It achieves this by providing

Indicative Summary
A report co-produced by UN, US, and EU agencies says that, if current policies are maintained, the ozone layer will be restored to 1980 values before the ozone hole appeared at different points in different places.

Informative Summary
An international agreement in 1987 to stop using the harmful chemicals that were damaging the layer has been successful, a major assessment says. A gaping hole in the layer was discovered by scientists in 1985. Just two years later, the Montreal Protocol was signed with 46 countries promising to phase out the harmful chemicals. The deal later became the first UN treaty to achieve universal ratification, and almost 99% of banned ozone-depleting substances have now been phased out. A report co-produced by UN, US, and EU agencies says that, if current policies are maintained, the ozone layer will be restored to 1980 values before the ozone hole appeared at different points in different places: 2066 over the Antarctic, 2045 over the Arctic, about in about two decades' time everywhere else.

Figure 2.1: An example of an indicative and informative summary of a BBC article (link)

an information searcher with a non-exhaustive summary allowing them to *screen* documents instead of reading them in their entirety. This increases productivity by reducing reading time and empowers the searcher to review more documents than before. This application was studied by Mani et al. (2002), where test participants were asked to find information in a collection of documents. They concluded the following:

> *Summaries as short as 17% of full-text length sped up decision-making by almost a factor of 2 with no statistically significant degradation in accuracy.*

The goal of an *informative summary* is to act as a *substitute* for the original document by exhaustively summarizing the source document. It is a self-contained summary that conveys all the information in the source document, therefore, removing the need for the source document at its original length. This similarly reduces the time spent and increases productivity, but demands a higher level of precision for users to accept a summary.

Figure 2.1 displays an example of both an indicative and informative summary. The indicative summary leaves out multiple key points, while the informative summary correctly conveys all the relevant information of the source document.

Early work on text summarization expressed an explicit focus on developing systems that produce *indicative summaries*, however, recent research is less clear on the intention. Both goals are related to the situational needs of the target user. Following the terminology of Nenkova et al. (2011), a summary can be assigned with a *summary type*, describing the style and intent of a summary. Summary types are non-exclusive and a summary can belong to multiple summary types. The summary types are:

Extractive versus Abstractive An extractive summary is an *excerpt* of the source document, which is constructed by selecting sentences in the source document and concatenating them into a summary. An abstractive summary is a free-text abstract of the document that, unlike an extractive summary, is unconstrained by the phrasing and writing style of the source document.

Single versus Multiple Document A summary that summarizes one document (news, lecture, paper) is called a *single document summary*, while one that summarizes multiple documents (reviews, complaints) is called a *multi-document summary*.

Generic versus Query-focused The relevant information in a document may vary depending on different readers' needs. To account for this a summary can either be a *generic summary* or a *query-focused summary*. A query-focused summary allows the reader to influence the summary by expressing a specific interest a priori through a *query*.

Mono-lingual versus Cross-lingual In the increasingly globalized societies of today, there is a need for cross-lingual summaries that convey information across language barriers. To account for this dimension, a summary that is written in the same language as the source document is called a *monolingual summary*, while a summary that is written in a different language different from the source is called a *cross-lingual summary*.

Type	**(a)**	**(b)**
(a) Generic / (b) Query-focused	X	-
(a) Extractive / (b) Abstractive	X	X
(a) Single / (b) Multi-document	X	-
(a) Mono / (b) Cross-lingual	X	X

Table 2.1: Summary types contributed to in this thesis.

Research on text summarization implies contributing to one of the above summary types, and covering all of them is well outside the scope of a single thesis. Specifically, this thesis contributes to the advancements of *single document, extractive, abstractive, monolingual, and cross-lingual summarization*. It does not explore multi-document or query-focused summarization (depicted in Table 2.1).

Task Formulation

The concept of a summary is often described as text that *summarizes the important information in a document*. However, specifics are often omitted, relying on descriptive but unquantifiable terms such as *important* or *salient* content. Although a strict formulation remains an open research question, this section provides a task formulation that defines the task in a more structured manner.

To define the summarization task let D denote a source text document, and S denote a text summary. D and S are sequences of words, w_i, and have lengths n and m, respectively ($|D| = n$, $|S| = m$).

$$D = [w_1, \ldots, w_n], \quad S = [w_1, \ldots, w_m] \tag{2.1}$$

The goal of developing a summarization system is to design a function f that maps D to a summary S such that S fulfills a desired set of properties (more on this below). It is central that any D has several or likely many adequate summaries, and the task is, therefore, not to produce an optimal summary, but rather, to produce one useful summary out of many summaries.

$$S = f(D) \tag{2.2}$$

The desirable properties of S depend on the target summary type and rely in particular on the meaning of *importance* for the specific input document. For the sake of generality and to cover all summary types two properties are desirable for any candidate summary:

1. *S is strictly shorter than D.*
2. *S conveys the relevant information that the user was looking for in D.*

These properties can be reformulated as constraints. The first is trivially quantified by comparing differences in lengths between S and D and can be expressed as $|S| \ll |D|$. To quantify the constraint in a continuous measure, a compression rate can be defined as:

$$\text{compression}(D, S) = 1 - \frac{|S|}{|D|} \tag{2.3}$$

The second property is less trivial to quantify and is closely related to the notion of *salience* (Conklin and McDonald, 1982). Successfully modeling this property plays a central role in modeling text summarization. Finding a strict definition, however, remains an active research topic to this day. With the lack standard definition of this property, this thesis provides the following:

Let $i(T)$ denote a saliency function that takes a text, T, as input and returns the salient information contained in T effectively capturing, *relevant information that the user was looking for*. This could in practice be n-grams, a distribution over words, or raw text. Using this function a summary's relevance can be determined by comparing it with $i(D)$. Measuring the overlap between S and $i(D)$ provides a concrete goal for a summarization system. This concept is visualized as a Venn diagram in Figure 2.2.

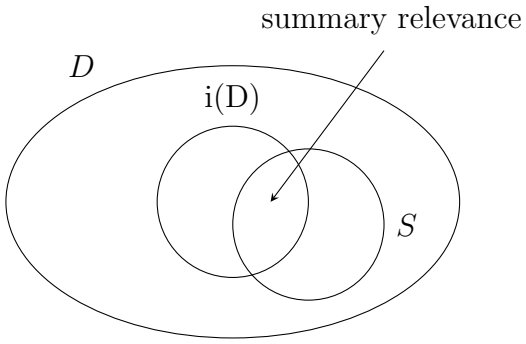


Figure 2.2: Venn diagram of a document, D, the salient content, $i(D)$, and a summary, S. $i(D) \cap S$ holds the shared information between the summary and salient information of D.

The goal of a summarizer is to maximize the overlap between $i(D)$ and S, (visualized by $i(D) \cap S$), which produces a summary that overlaps as much as possible with the important information contained in D. Assuming access to a similarity function, $\text{Sim}(t_1, t_2)$, that returns the overlap between two texts,

the relevance of a summary can be expressed as:

$$\text{relevance}(D, S) = \text{Sim}(i(D), S) \tag{2.4}$$

In practice, neither $i(T)$ nor $\text{Sim}(t_1, t_2)$ is generally available and designing reasonable implementations remains a central research question in summarization research.

Peyrard (2019) conceptualizes a possible answer with an information-theoretical motivated framework. In Peyrard's work, *relevance* is modeled probabilistically by measuring how well a summary approximates the information of the source document. From an information-theoretical perspective, this means that observing S should reduce any uncertainty about the information contained in D. This invites parallels to the measure of cross-entropy of two random variables, defined as:

$$\text{H}(X, Y) = -\sum_i p(x_i) \log p(y_i) \tag{2.5}$$

With probability distributions over the words[1], of the source document and the summary, P_D and P_S, Peyrard defines relevance as the cross-entropy between the word distributions in D and S:

$$\text{relevance}(S, D) = \sum_w P_S(w_i) \log P_D(w_i) \tag{2.6}$$

From this, it follows that if S contains the same information as D then S is a relevant summary, and at least conceptually, a perfect summary system would produce a summary S such that $\text{relevance}(S, D) \approx \text{relevance}(D, D)$ which is equivalent to the entropy of the source document $\text{H}(D)$.

$$\begin{aligned} \text{relevance}(D, D) &= H(D, D) \\ &= \sum P_D(w_i) \log P_D(w_i) \\ &= H(D) \end{aligned} \tag{2.7}$$

Peyrard's formulation of relevance applies primarily to the notion of *generic summarization*, and applying this to other summary types would demand extending the expression. For example, query-focused summarization would likely require introducing a query, q, to allow focusing on particular aspects of a source document. To incorporate this into Peyrard's framework, relevance

[1]This could be term frequencies or probabilities from a unigram language model.

would need to be rewritten to include q[2]:

$$\text{query-focused relevance}(S, D, q) = \sum_{w_i} P_S(w_i) \log P_D(w_i|q) \tag{2.8}$$

Relevance provides a goal property of a summary, however, it is important to avoid optimizing only for relevance as it aligns poorly with the summarization task. For example, relevance can be maximized by not removing any information and simply assigning $S = D$, thus, not reducing the amount of text. Likewise, compression can be maximized by producing an empty string and assigning $S = \emptyset$, thus, not conveying any information. Balancing these two constraints is what lies at the core of the summarization task and a summarizer must walk the line between producing summaries that *remove text while retaining relevance*.

2.2 Evaluating a Summary

Evaluating a summary and by extension, the performance of a summarization system remains an open research topic (Bosselut et al., 2021; Fabbri et al., 2021). The goal of evaluating a summary is to determine to which degree it satisfies the constraints described in the previous section, namely, being shorter than the source document, and conveying relevant information. To operationalize this, it is common to determine to which degree the information in the summary *overlaps* with the *important information* of the source document. To capture this notion, let $\mathcal{M}(D, S)$ denote an evaluation metric that takes a document and a summary, and returns an evaluation score. A straightforward approach to this is to have humans rate the summary.

$$\mathcal{M}(D, S) = \text{human judgement}$$

This implicitly tasks the human with identifying the important information in D, meaning that part of the evaluation process includes summarizing the document D to evaluate S. Since humans are too costly for most practical settings, it is common to approximate human preference with automatic methods. This introduces an immediate problem. Since humans must summarize, to evaluate how can an automatic method evaluate without a summarizer? To circumvent this recursive conundrum, it is common practice to assume that important information in D is captured by reference summary

[2] A suggestion made by this thesis

R, such that $R \approx i(D)$. This changes the signature of $\mathcal{M}$ from an evaluation that directly compares the document and the summary $\mathcal{M}(D, S)$ to comparing the summary and the reference summary $\mathcal{M}(R, S)$. All automatic evaluation metrics for summarization compare summaries with references and do not directly consider the contents of the source document.

ROUGE

The most common automatic evaluation metric and the standard for text summarization is ROUGE (Lin, 2004). ROUGE is an n-gram-based metric that computes the similarity between a summary and a reference summary. The similarity here refers to the n-gram overlap between the summary and reference summary.

$$\text{ROUGE-n}(R, S) = \frac{|grams_n(R) \cap grams_n(S)|}{|grams_n(R)|} \tag{2.9}$$

ROUGE defaults to normalizing by the number of n-grams in the reference and is, therefore, a kind of "n-gram recall", parameterized by n. This produces a bounded score in the range $[0, 1]$ that measures the degree of n-gram overlap between R and S. The metric rewards a summary that contains n-grams that are also contained in R but does not penalize non-overlapping n-grams. To capture notions of both information overlap and text fluency it is common practice to report several variations of ROUGE with different parameters, $n = \{1, 2, L\}$ where L refers to the longest common subsequence of n-grams (Cormen et al., 2022). To exemplify the metric, let a summary, S, and a reference summary, R be instantiated with the following texts:

$$R = \text{I am happy}$$
$$S = \text{I am so happy}$$

Computing ROUGE-{1,2,L} for this pair produces the following scores:

$$\text{ROUGE-1}(R, S) = \frac{3}{3}, \quad \text{ROUGE-2}(R, S) \quad = \frac{1}{2}, \quad \text{ROUGE-L}(R, S) = \frac{1}{1}$$

ROUGE-1 evaluates to $\frac{3}{3}$ because all unigrams in the summary, $grams_1(S) =$ {"I", "am", "happy" }, are also in the reference summary, $grams_1(R) =$ {"I", "am", "so", "happy" }. ROUGE-2 is obtained in the same way but by instead using bigrams and ROUGE-L is computed by finding the longest common (non-contiguous) n-gram and dividing its length by the number of unigrams

in the reference. Notice that ROUGE does not penalize novel n-grams (such as the unigram "so" for ROUGE-1). This means that a system can game the ROUGE metric by including additional n-grams without consequence. ROUGE has since its original introduction been modified to better fit more generative text models. Instead of the recall metric it was initially intended to be, it is now common to report the harmonic mean (F1-score) between ROUGE_{recall} and $\text{ROUGE}_{precision}$.

$$\text{ROUGE-n}_{recall}(R, S) = \frac{|grams_n(R) \cap grams_n(S)|}{|grams_n(R)|} \tag{2.10}$$

$$\text{ROUGE-n}_{precision}(R, S) = \frac{|grams_n(R) \cap grams_n(S)|}{|grams_n(S)|} \tag{2.11}$$

$$\text{ROUGE-n}_{f1}(R, S) = 2\frac{\text{ROUGE-n}_{precision} \cdot \text{ROUGE-n}_{recall}}{\text{ROUGE-n}_{precision} + \text{ROUGE-n}_{recall}} \tag{2.12}$$

This was introduced to allow text generation models which can not easily control length to compete (Chopra et al., 2016; Nallapati et al., 2016). This alteration has been rather undisputed by the research community although it remains unclear how well-motivated it is.

Alternative Methods

An immediate limitation of ROUGE is its' inability to capture language variation. Even small changes to a summary can lead to vastly different scores despite being semantically equivalent. For example in the above example, if the words "I am" are replaced with the contraction "I'm", ROUGE-1 drops to 0.3, and R2 to 0. This is a consequence of the sparse nature of language, and can only directly be resolved by expanding R to include multiple paraphrases. To address this, methods based on learned text representations (Mikolov et al., 2013; Pennington et al., 2014; Devlin et al., 2019) have been proposed to circumvent the shortcomings of n-grams and compute the similarity between semantic vectors. Popular measures are `MoverScore` (Zhao et al., 2019) and `BERTScore` (Zhang et al., 2019a). They provide a means to effectively close the sparsity gap. These measures similarly assess the similarity between S and R, but instead of hard-aligning n-grams methods, the similarity is relaxed to soft-alignment, matching words based on semantic similarity.

2.3 Neural Networks

At the center of most contemporary summarization systems lies a sufficiently large artificial neural network (NN). NNs are computational models that when configured in the right way can theoretically model *any function* (Hornik et al., 1989) which has shown to be incredibly effective at modeling language-related tasks. Paired with sufficiently large amounts of data, NNs have repetitively shown to successfully produce robust and accurate predictions, and have as a result become the standard modeling paradigm in NLP. Text summarization is no exception to this and summarization research has in the past 10 years fully embraced neural designs as the default approach to building summarization systems.

NNs are in part attractive due to being flexible computational models, allowing designs to be changed to accommodate task-specific needs. As a result, a plethora of NN architectures have over the years been proposed for NLP. This includes modeling n-grams with convolutional neural networks (Le Cun and Bengio, 1994), introducing sequence-level memory with long short-term memory networks (LSTMs) (Hochreiter and Schmidhuber, 1997), and capitalizing on the attention mechanisms of the transformer model (Vaswani et al., 2017).

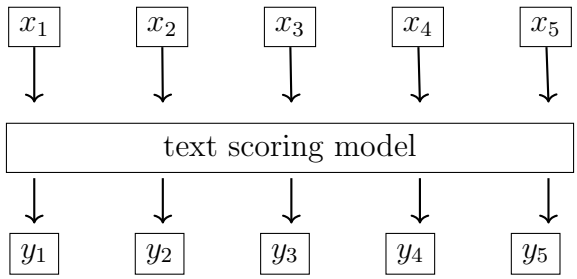
Common for all of such architectures is their ability to model *sequences*. For NLP, a NN is a parameterized function, f_θ, that takes a text sequence as input, $x = [x_1, \ldots, x_n]$, and returns a prediction, $\hat{y}$, suitable for the task at hand.

$$\hat{y} = f_\theta([x_1, \ldots, x_n]) \tag{2.13}$$

However, the shape of the output prediction, $\hat{y}$, varies widely across tasks, ranging from a single binary value to long texts. For text summarization, $\hat{y}$ generally takes two forms, either being *scores over the input text* or a *newly generated text sequence*.

2.3.1 Models that Score Text

NNs that score text belong to a category of models that directly score the input sequence. Such models can return scores for each element in the input sequence, a subset of elements, a subsequence of elements, or even a score for the entire input sequence. Such models in NLP are sometimes referred to as *discriminators* and are designed to attribute scores to input like text, words, or sentences. The ability to assign scores to, in particular *sentences*, makes text scoring models relevant to summarization systems that target *extractive summaries* which will be further described in Section 2.5.1. More generally, these models return a number of predictions that are fixed by the number of elements in the input sequence.



This type of model translates well to tasks that involve conducting text analysis like named entity recognition (Wang et al., 2021a), part-of-speech tagging (Wang et al., 2021b), and extractive summarization (Liu and Lapata, 2019). It is, however, less suited for tasks that require more flexible output predictions and are not directly linked to explicit elements. Such tasks are tasks that deal with text generation for which a different type of model is more appropriate.

2.3.2 Models that Generate Text

NNs that are capable of generating text belong to a category of models called *sequence-to-sequence* models, with summarizers often implementing a variation called *encoder-decoder* models (Sutskever et al., 2014). With encoder-decoder-type architectures, NNs could conditionally produce text from existing text. The design is simple, but effective and consists of two parts; an *encoder* and a *decoder* (hence the name). Such models work by encod-

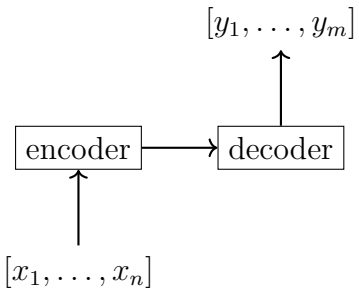


Figure 2.3: The architecture of an encoder-decoder model.

ing the source text, $X = [x_1, \ldots, x_n]$, using the *encoder*, and then feeding the encoded input to the decoder, which subsequently generates a new text, $\hat{y} = [y_1, \ldots, y_m]$. Notice here the different lengths of the input and the output, $n \neq m$. For summarization, this translates nicely to the procedure of encoding a source document, $D = [x_1, \ldots, x_n]$, and directly mapping it to a concise, but informative summary, $S = [y_1, \ldots, y_m]$:

$$S = f_\theta(D)$$
$$[y_1, \ldots, y_m] = f_\theta([x_1, \ldots, x_n])$$

An attractive trait of this kind of model is its ability to output sequences that are not directly tied to the length of the input sequence. This provides a both flexible and expressive model that allows producing sequences of variable length. This ability to generate variable-length outputs is a property that aligns well with tasks that inherently require generating text. Examples of this are machine translation (Vaswani et al., 2017), abstractive summarization (See et al., 2017), abstractive question answering Sun et al. (2019), and digital dialogue agents (Tyen et al., 2022). This makes encoder-decoder models well-suited for summarization systems that target *abstractive summaries*, and in fact, constitutes an entire paradigm of summarization systems. This is introduced in Section 2.5.2.

2.3.3 Pre-trained Language Models

Perhaps the strongest tool available to an NLP practitioner is a pre-trained language model (PLM). PLMs are large neural networks trained on large amounts of text data. Using a *language modeling objective* they are optimized to estimate the probability of words given their surrounding context, motivated by *distributional semantics* (Firth, 1957). This produces models that can generate fluent, coherent, and generally high-quality text and provide robust text representations which increase the accuracy and robustness of systems that they are incorporated into.

Model	**Training Data**
BERT (Devlin et al., 2019)	16GB
RoBERTa (Liu et al., 2019)	160GB
XLNet (Yang et al., 2019)	158GB
BART (Lewis et al., 2020)	160GB
T5 (Raffel et al., 2020)	745GB

Table 2.2: Commonly used pre-trained language models paired with the size of the training set they are trained on.

Without exception, all proposed summarization systems in the past few years build on top of an existing PLM. The best practice is, therefore, to load a publicly available PLM, optionally extend or modify its architecture, before then fine-tuning it on summarization data (depicted in Figure 2.4). Systems that improve upon this recipe (see Section 2.5.1 and Section 2.5.2) have only recently been suggested, however, these systems make improvements primarily by boosting the performance of said PLM recipe.

The effectiveness of PLMs can primarily be attributed to the text data they have been trained on, and this is generally not a small amount of data. Table 2.2 lists well-established PLMs and the amount of text data they have been trained on, showing that contemporary PLMs require *lots of data*. This highlights the necessity of ensuring access to large amounts of unlabeled text, as large parts of modern summarizers rely on the availability of a PLM.

Figure 2.4: A common pipeline to training a summarizer

2.4 Required Data

Summarization systems are increasingly reliant on data. Whether it is the raw text used to train pre-trained language models or paired document-summary pairs to fine-tune summarizers, the reality is that today's systems are not viable without lots of data. It is, therefore, essential to ensure access to sufficient data to build competitive summarization systems for any language. This section provides an overview of currently available text datasets and a comprehensive overview of summarization datasets.

2.4.1 Text Datasets

Common approaches to acquiring large amounts of text data usually involve filtering large snapshots of the internet distributed by online archives like CommonCrawl[3]. Such approaches provide virtually endless amounts of text data with billions of web pages publicly available. To facilitate the necessary resources to create PLM a multitude of datasets have been created, each new dataset larger than the previous. The BookCorpus (Zhu et al., 2015) is a large collection of English 11,000 books (4.6 GB), OSCAR (Ortiz Suárez et al., 2019; Abadji et al., 2022) compiles text from Wikipedia for 152 languages (<0.1 GB to 1.2 GB), C4 (Raffel et al., 2020) cleans all English data in CommonCrawl (305 GB), mC4 extends C4 this to 108 languages (0.15 GB to 301 GB), and The Pile (Gao et al., 2020) compiles text from 22 English datasets (825 GB).

These datasets are the data foundation that has powered influential mono- and multilingual models. While it is undeniable that access to this data has proven to be useful, a closer look at the datasets challenges the premise of the abundance of text data for all languages. Inspecting CommonCrawl[4] reveals that 149 of the 161 (92%) languages represent less 1% of the content, and 123 languages (76%) represent less than 0.1%. This contradicts the common belief that text data is an abundant resource and goes to show that statement is a feature reserved primarily for a few privileged languages. Recent research has further pointed out several challenges related to text datasets collected from the web. Kreutzer et al. (2022) studied 205 language-specific datasets derived from online sources and found serious issues with several datasets. They found that *at least 15* of the distributed language datasets did not contain *any useful data* and that multiple datasets contained text data of

[3] `https://commoncrawl.org/`

[4] `https://commoncrawl.github.io/cc-crawl-statistics/plots/languages`

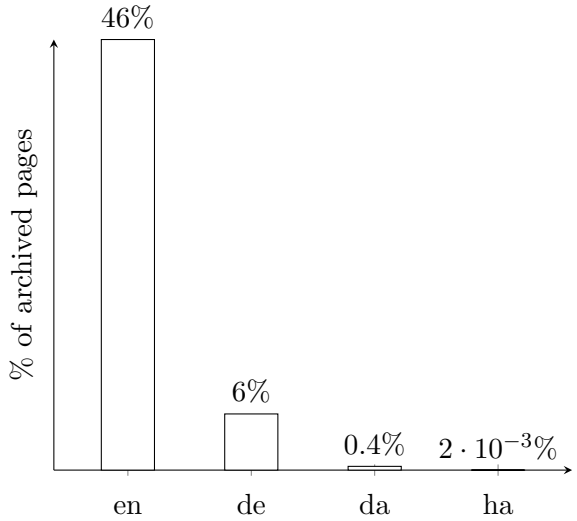


Figure 2.5: Distribution of archived web documents contained in Common Crawl. English constitutes almost half of all collected documents. De, da, and ha are German, Danish and Hausa, respectively.

such poor quality that it was practically useless. Haas and Derczynski (2021) documented a similar phenomenon for the nordic languages, reporting that text was consistently labeled with the wrong language due to errors made by automatic language identifiers, ultimately causing Norwegian Bokmål to end up in the Danish dataset partitions.

2.4.2 Summarization Datasets

While text data and PLMs have become a core part of summarizers, labeled summarization datasets remain a necessity to develop summarization systems. Summarization datasets, $\mathcal{D}$, are collections of tuples of texts, each containing a document, D, and one or several reference summaries, R.

$$\mathcal{D} = \{(D_i, R_i)\}_{i=1}^{n} \tag{2.14}$$

State-of-the-art summarization systems require massive amounts of training data, often containing hundreds of thousands of paired document-summary pairs. Manually creating datasets of such sizes is an expensive endeavor. Nguyen and Daumé III (2019) reported paying an average of 1.5 US dollars per sample, while more recently Wang et al. (2022) reported 6 USD dollars, putting the starting price of a common summarization dataset somewhere between 300.000 and 2 million USD. In comparison, Ishita et al. (2020) reported

paying just 0.3 USD per labeled sample when creating a text classification dataset. As a result, it is common practice to create datasets by searching for data that can reasonably be repurposed as summarization data.

Common Summarization Dataset

Dataset	Size	Publication	Citations
Gigaword	4×10^6	Rush et al. (2015)	2200
CNN & Dailymail	3.1×10^5	Nallapati et al. (2017)	3680
XSum	2.2×10^5	Narayan et al. (2018a)	295
Newsroom	1.3×10^6	Grusky et al. (2018)	209

Table 2.3: English summarization datasets. Citation counts are collected from Semantic Scholar

Gigaword With the lack of a sufficiently large summarization dataset, Rush et al. (2015) repurposed the Gigaword corpus (Graff et al., 2003)[5] as a summarization dataset. The dataset is a collection of over 4 million English news articles from the Associated Press and the New York Times and was developed as a linguistic resource for corpus statistics and language modeling. Since each article was annotated with a title the authors proposed recasting titles as summaries. Although a title is not directly a summary, the paper showed that neural networks could be trained to generate descriptive titles which served the dual role of a summary.

CNN & Dailymail The question-answering dataset CNN & Dailymail (Hermann et al., 2015) was in a similar fashion repurposed for summarization by Nallapati et al. (2016). The English dataset is comprised of ∼312,000 news articles from the US-based news outlet CNN, and the British tabloid *The Daily Mail*. Unlike the titles in Gigaword, this dataset leverages article "highlights" that contain facts related to the article. Although the highlights were editorially intended as "fact boxes", they have been broadly accepted as summaries by the community and the dataset stands as *the standard* summarization benchmark dataset.

[5] `https://catalog.ldc.upenn.edu/LDC2003T05`

Dataset	Size	Publication	#L	Citations
LCSTS	2.4×10^6	(Hu et al., 2015)	1	242
GlobalVoices	7.5×10^4	(Nguyen and Daumé III, 2019)	15	18
WikiLingua	7.6×10^5	(Ladhak et al., 2020)	18	74
Liputan6	2×10^5	(Koto et al., 2020)	1	17
MLSum	1.5×10^6	(Scialom et al., 2020)	5	64

Table 2.4: Non-English Summariation Datasets. #L denotes the number of languages covered by the dataset. Citation counts are collected from Semantic Scholar.

XSum Narayan et al. (2018a) adopted the same strategy of leveraging website layouts to extract summaries and targeted the British news outlet BBC. This produced the English dataset *XSum* which consists of ∼225,000 article-summary pairs. Instead of fact boxes, each article starts with a bolded introduction paragraph. The effectiveness of this methodology has been questioned as it is unclear whether the summaries are possible to reconstruct from the remaining article body as the information contained in the extracted introduction paragraph is not necessarily to be repeated in the rest of the article (Cao et al., 2022).

Newsroom Common for previous datasets is that they rely on the website-specific layout to retrieve summaries. The Newsroom dataset (Grusky et al., 2018) avoids this constraint by introducing a method that allows extracting a summary from an arbitrary news site. Instead of relying on consistent site layouts, a summary is extracted from meta-data[6] published with the article. It is used to control what is displayed when the article is shared on social media. The meta-data includes lots of information, including a *a description* which usually functions as a summary. Applying this to a large list of news outlets is what has produced the largest English summarization dataset to date, consisting of 1.3 million article-summary pairs collected from 38 different online news outlets.

Non-English Datasets

The attentive reader will notice that the previous datasets cover only one language, English. One may be inclined to attribute this to cherry-picking, however, the reality is that summarization data for non-English languages are long and far apart. This section gives a brief overview of existing large

[6] `https://ogp.me`

non-English mono-and multilingual summarization datasets.

LCSTS (Chinese) The perhaps most influential non-English summarization dataset is the *LCSTS* dataset (Hu et al., 2015). It is a Chinese dataset containing ∼2.4 million blog posts, each paired with a summary. The dataset was built using data from the microblogging website, *Sina Weibo*, and takes advantage of highlighted summaries, similar to that of XSum. The dataset is the largest and most cited non-English dataset to date.

Liputan6 (Indonesian) Koto et al. (2020) built a large-scale summarization dataset based on the content from the news outlet *liputan6.com*. The dataset consists of ∼215,000 document-summary pairs in the Indonesian language and is created by extracting summaries embedded in the JavaScript code of the webpage article.

GlobalVoices (15 Languages) Nguyen and Daumé III (2019) was the first to collect data for multiple languages from a multilingual website. Using data from *globalvoices.org*, a volunteer-based news outlet that publishes news articles in 54 languages, they built the first multilingual dataset named *GlobalVoices*. The dataset contains a total of ∼75,000 article-summary pairs across 15 languages and is distributed with an additional subset set of manually written summaries obtained through crowd-sourcing. The dataset is highly unbalanced across languages ranging from less than 100 to 500 samples per language. Also, the quality varies with primarily the manually written summaries being of enough quality for practical use.

WikiLingua (18 Languages) Ladhak et al. (2020) introduced the first large cross-lingual dataset *Wikilingua* covering 18 languages. It was built with data from the website Wikihow[7], an online wiki containing *how-to* articles. Each article contains a series of instructions split into steps. The dataset is created by extracting the lead text of each step and concatenating them into a summary. Similar to GlobalVoices, the dataset is highly unbalanced with languages ranging from 4,500 to 141,000 samples.

MLSum (5 Languages) The latest addition of multilingual datasets is the MLSum dataset (Scialom et al., 2020). This is a multilingual summarization dataset covering five high-resource languages: French, German, Spanish,

[7] `https://www.wikihow.com`

Russian, and Turkish. The dataset contains a total of 1.5 million samples balanced equally over the languages. This work was conducted concurrently with the multilingual dataset published as part of this thesis (Section 3.3) and similarly uses website meta-data to extract summaries. To build the dataset the authors select one news outlet for each language to collect from.

2.5 Summarization Systems

With this thesis contributing to extractive, abstractive and cross-lingual summarization, this section gives a quick introduction to state-of-the-art systems within each respective paradigm.

2.5.1 Extractive Summarization

A simple way to create a summary is by combining extracts of the input document (Edmundson, 1969). This represents the earliest approach to modeling the summarization task and is called *extractive summarization*. With text generation early on being more difficult compared to now, summarizing *extractively* had the convenient side-effect of delegating the task of "writing" to the source document. Systems that adopt this approach build on the notion that sentences can serve directly as a summary and frame summarization as a *text scoring task* (as introduced in Section 2.3.1).

To define extractive summarization let a source document that is composed of n sentences, instead of words be denoted by $D = [x_1, \ldots, x_n]$. Let $\mathcal{C}(D)$ then denote a function that takes a document and returns all possible combinations of sentences, also called the *candidiate summary set*. Because an extractive system is restricted to considering combinations of sentences in D, this can be formulated as a subset of the power set of sentences $x_i \in D$.

$$\mathcal{C}(D) = \{C \in \mathcal{P}(D)\} \tag{2.15}$$

To allow specifying a desired summary length let $\mathcal{C}_k(D)$ denote the candidate summary space of summaries which are composed of k sentences:

$$\mathcal{C}_k(D) = \{C \in \mathcal{P}(D) \,|\, |C| = k\} \tag{2.16}$$

The size of this set is large, and is expressed by $\frac{n!}{(n-k)!}$. Since different permutations of sentences do not convey additional information (but do influence

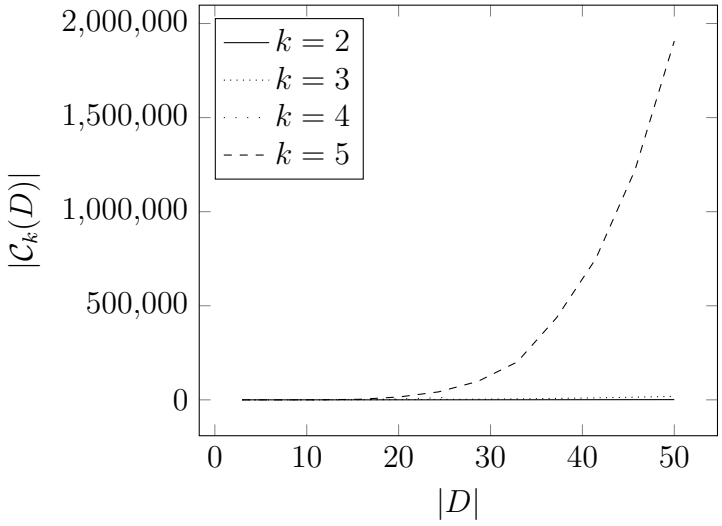


Figure 2.6: Size of different candidate summary spaces with fixed summary length, k as a function of document length.

discourse and fluency), this quantity can in practice be relaxed to the binomial coefficient:

$$|\mathcal{C}_k(D)| = \binom{|D|}{k}$$

Figure 2.6 plots the size of the candidate spaces $\mathcal{C}_k(D)$ as a function of document length with different summary lengths $k = [2, 3, 4, 5]$. Since it is natural to consider summaries of different lengths, it is clear that the sheer number of candidate summaries makes it intractable to evaluate the entire space in most practical settings. For example, given a document, D', of length 40 (sentences), there are nearly 10,000 candidate summaries containing three sentences, and adding another sentence expands the space more than tenfold.

$$|\mathcal{C}_3(D')| = 9,880$$
$$|\mathcal{C}_3(D')| + |\mathcal{C}_4(D')| = 101,270$$

To avoid dealing with an intractable number of candidates, it is common practice for extractive systems to make approximations of the candidate space. This means avoiding scoring combinations of sentences (higher-order scoring) and instead scoring sentences independently (first-order scoring).

This reduces the task to sentence classification and significantly simplifies the computational complexity of a model, although it reduces its expressiveness. Under this approximated formulation, an extractive system obtains a summary by scoring each sentence in D, producing $\hat{Y} = [\hat{y}_1, \ldots, \hat{y}_n]$. These

	$\lvert D\rvert$	$\lvert\mathcal{C}_k(D)\rvert$
candidate space	n	$\frac{n!}{k!(n-k)!}$
approximation	n	n

Table 2.5: Size of the entire summary candidate space versus it's first-order approximation. These quantities are a function of the document length, $|D|$, and summary length, k.

scores are then used to select m sentences such that $m \ll n$, and concatenated into a summary S. This formulation is effective but poses a challenge for models based on supervised learning as it creates a misalignment between the type of labels a classifier needs, and the labels that are found in summarization datasets. Specifically, summarization datasets contain labels that are *plain text*, not binary sentence labels.

To obtain sentence labels, y, it is common practice to utilize an *oracle model*. An oracle model is an extractive summarizer with access to both the gold reference R and the evaluation metric $\mathcal{M}$. Thus, the oracle is an "all-knowing" model that provides a means to predict the optimal solution, since it can directly maximize $\mathcal{M}$.

$$y = \mathbb{1}(x_i \in c^*) \,\text{where}\, c^* = \underset{c \in \mathcal{C}(D)}{\arg\max}\, \mathcal{M}(c, R) \tag{2.17}$$

Labels are practically obtained by scoring the candidate space and assigning positive labels to sentences that are part of the highest-scoring candidate summary. With $\mathcal{C}_k(D)$ being intractable to enumerate it is necessary to approximate the space with an efficient search strategy such as greedy search which selects one sentence at a time (Nallapati et al., 2017). Greedy search is often enough to find the global optimum (Xu and Lapata, 2023). While this provides a means of obtaining extractive sentence labels, it does not necessarily produce fluent or even relevant summaries. Figure 2.7 depicts such a case where the oracle summary is of poor quality, exhibiting poor coherence, and leaving discourse makers unconnected.

First Order Extractive Systems

Historically, extractive systems have modeled summarization using the first-order approximation, producing summaries by individually scoring sentences and concatenating the highest-scoring sentences into a summary. The num-

Reference Summary
Japanese football team Vegalta Sendai had best ever season last year. Finished fourth in Japan's J-League. Team's stadium was ruined by last year's earthquake. Team inspired to go on two 11 match unbeaten runs.

Oracle Summary
Yet in March last year Vegalta Sendai was looking forward with optimism to a rare season in Japan's top flight. But their fourth placed finish represented their best ever season. The team's stadium was declared to be "in ruins" by J-League chairman Kazumi Ohigashi, its training ground destroyed.

Figure 2.7: Reference summary and corresponding (greedy) oracle summary sampled from the CNN/Dailymail dataset.

ber of considered sentences, k, is a parameter of such models and is usually optimized for a target dataset.

$$S = \text{concat}(\Big\{ x_i \in D \mid f(x_i) \in \text{top}\left(\{f(x_i)\}_i^n, k\right) \Big\}) \tag{2.18}$$

Because first-order inference forces models to make independent predictions with incomplete knowledge, most progress on extractive systems has revolved around injecting global information into the sentence representations, thus allowing the model to make more informed predictions. For early approaches, this meant designing handcrafted feature extractors relying on lexica, document and corpus statistics (Mihalcea and Tarau, 2004; Erkan and Radev, 2004; Gillick et al., 2008). With neural networks becoming the norm, so has sentence representations shifted to neural sentence representations learned from data. Also here, research has focused on how to get neural networks to inject more document-level information into sentence representations.

Cheng and Lapata (2016) encoded sentences hierarchically using long short-term memory networks (Hochreiter and Schmidhuber, 1997) and found that autoregressive inference was an effective approach. Narayan et al. (2018b) investigated a similar architecture but instead proposed an alternative loss based on reinforcement learning and found that optimizing the system using a summary-level signal improved performance. Zhou et al. (2018) confirmed this in a related study, also finding that summary level resulted in performance improvements. With the recent successes of fine-tuning transformers-based PLMs, Liu and Lapata (2019) proposed an extractive system built on top of BERT (Devlin et al., 2019). With the BERT model's ability to contextually encode an entire document, the authors showed that by simply adding additional classification tokens for each sentence, a PLM could be effectively

repurposed as a performant extractive summarization system.

Higher Order Extractive Systems

First-order approximations have limited expressiveness and recent research has suggested that the limits of first-order models have been reached.

Zhong et al. (2020) argue that there is a need for a *paradigm shift* to further improve extractive summarization system designs. They explore this by conceptualizing extractive summarization as a *semantic matching problem*. Specifically, the relevance of a summary can be established by measuring the semantic similarity between the *source document* and *the candidate summary*. A summary is produced by computing the similarity between the source document, D, and the elements of the candidate summary set, $\mathcal{C}(D)$, selecting the candidate summary that is the most similar to the source document. However, as established, $\mathcal{C}(D)$ is not tractable to enumerate, and the method does not resolve the combinatorics. Instead, the method relies on heavily pruning $\mathcal{C}(D)$. The pruned candidate space is obtained by running a first-order model, g_θ, to produce a filtered document D' that contains the top 5 sentences as scored by g_θ.

$$D' = [x_i \in D \,|\, x_i \in \text{top}(\{g_\theta(x_i)\}_i^n, 5)] \tag{2.19}$$

With the length of the input document being fixed to a small sentence count the candidate summary space becomes feasible to exhaustively score, with values for k either producing 5 or 10 candidate summaries:

$$\mathcal{C}_{\{1,5\}}(D') = 5$$
$$\mathcal{C}_{\{2,3,4\}}(D') = 10$$

The method is implemented using a siamese network (Bromley et al., 1993) that embeds D and elements of $\mathcal{C}(D')$ into a shared vector space and computes similarity using cosine similarity.

$$S = \underset{c \in \mathcal{C}(D)'}{\arg\max} \cos(f_\theta(D), f_\theta(c)) \tag{2.20}$$

This approach was shown to be effective, providing sizable improvements over previous systems and showed that increased expressiveness of models allows better performance. This makes the approach a pipeline method, boosting the performance of g_θ at the cost of additional parameters and computational

costs. However, since the model is not self-contained, it is reasonable to question whether doubling the model size for reranking is justified.

A different approach to higher-order scoring is to accept that enumerating $\mathcal{C}(D)$ blindly is intractable, and rather explore it using an informed strategy. Narayan et al. (2020) addressed the combinatorial challenge by introducing a search strategy over sentences. Instead of tackling the entire candidate summary space, a *searching for a summary* over the candidate summary space. This is achieved by iteratively appending a sentence to a *partial summary* until a summary has a desired length. Each step, t, is defined by:

$$S'_t = \text{concat}(S'_{t-1}, \hat{S}'_t) \tag{2.21}$$
$$\text{where } \hat{S}'_t = \arg\max_{x_i \in D} f_\theta(x_i, S'_{t-1})$$

Where S'_t is a partial summary at step t. This approach is similar to the way an extractive oracle obtains sentence labels, and the way text generation models generate text (but selecting a *sentence* instead of a *word* at each step). Searching for a summary instead of enumerating potential candidates is a promising research direction that directly addresses the daunting combinatorics of $\mathcal{C}(D)$, providing an effective method to model high-order extractive summarization. The results showed to be particularly effective when applied to *data-to-text*, a type of data that requires a higher level of planning. Furthermore, the results confirmed the conclusions of Zhong et al. (2020), namely, that higher-order modeling for extractive summarization is possible and provides better performance.

To implement this design a system was developed using recent advancements to the transformer architecture, namely, the extended transformer (Ainslie et al., 2020) the hierarchical transformer (Zhang et al., 2019b), which enabled the system to take advantage of the hierarchical nature of documents and longer contexts.

2.5.2 Abstractive Summarization

Summarization technology is useful not only because it gives faster access to information, but also because it allows access to information in a style preferable to the user. Extractive systems only satisfy this goal if the source document is written in the same style as the user desires. Far from all documents satisfy this constraint, and certain text domains do not exhibit this property at all. Such a domain could be dialogue transcripts, where the text document is derived from a conversation, consisting of sequences of often partially overlapping *utterances*. Summarizing such documents involves identifying key points, prioritizing information, and producing new text which can not be achieved by extracting sentences. An abstractive summary is likely more suitable for such settings as it is unbound by the wording and style of the source document. This allows *abstracting* over the contents of the document producing a summary that exhibits properties like information aggregation, compression and paraphrasing. Figure 2.8 contains an example conversation along with an extractive and abstractive summary, emphasizing the importance of matching documents and user needs.

To develop systems that allow this kind of *abstraction* a family of models exists under the name *abstractive summarization system*.

Abstractive Summarization Systems

Abstractive systems have in recent years gained immense traction, becoming the most actively researched system design[8]. Today, an abstractive system is generally implemented as a sequence-to-sequence model. Abstractive summarization systems are, therefore, *models that generate text* as described in Section 2.3.2. Abstractive summarizers are easy to implement because they can be trained directly on reference summaries. Unlike extractive systems that rely on oracle summaries, which can be awkward at times (see Figure 2.8), an abstractive system neither needs to restrict expressiveness nor use suboptimal training data, allowing for modeling summarization directly as a *text-to-text* task.

Rush et al. (2015) were the first to investigate a sequence-to-sequence system and explored different neural architectures. They implemented a bag-of-words system and found that attention was particularly effective in modeling summarization. Chopra et al. (2016) later implemented an attention-augmented recurrent neural network and found this to provide even further

[8] 4/5 papers published papers at ACL 2022 develop or analyze abstractive systems

A: Hello, this is xxx hotline. May I help you?
C: I've got an order saying that it has been delivered but I haven't received yet. When I checked it, it shows that the deal is done.
C: But err... I haven't received anything.
A: I got it. Then, could you please provide your username or the binding phone number of the application?
[. . .]
A: Did you place the order today?
C: Err... No, it was yesterday but he told me he would deliver it today. Hum, I checked the message in the morning but I haven't received anything.
A: Humm, ok. I see. I am gonna contact the deliveryman. Is that okay? I will check it for you and call you back later.
C: Ok, ok. That's good.

Abstractive Summary: The user called us because the order shows that it has been delivered but he did not receive it at all. I replied that I would check it by contacting the deliveryman.

Extractive Summary: **C**: I've got an order saying that it has been delivered but I haven't received yet. When I checked it, it shows that the deal is done. **A**: I am gonna contact the deliveryman. I will check it for you and call you back later.

Figure 2.8: An example of a conversation paired with an abstractive and extractive summary. The transcript and the abstractive summary originate from Zou et al. (2021) while the extractive summary is created for this figure.

improvements. A challenge faced by these models was that they struggled with modeling out-of-vocabulary words like named entities and topic-specific terms. To address this, Nallapati et al. (2016) proposed a switch mechanism that allowed the network to copy tokens from the input document, a concept that was generalized the following year by See et al. (2017) in the influential *pointer generator summarizer*. While this helped systems handle out-of-vocabulary problems, it caused systems to overeagerly copy long sequences from the input, producing repetitive loops which resulted in incoherent and verbose summaries. To resolve this Gehrmann et al. (2018) proposed a *bottom-up* design that incorporated an external *content-selector* that blocked the generation process from copying words that it deemed unlikely to appear in a summary.

Many of these challenges have been addressed by adopting innovations developed for machine translation and pre-trained language models. Techniques such as *subword tokens* (Sennrich et al., 2016a) largely removed out-

of-vocabulary problems and improved decoding techniques reduced models' tendencies to repeat themselves (Keskar et al., 2019). Furthermore, fine-tuning PLMs on summarization data has been shown to provide flat performance improvements (Radford et al., 2019; Lewis et al., 2019; Raffel et al., 2020), resulting in most contemporary abstractive systems being built in this manner. This shift has largely created a stalemate in system development focused on architectural solutions and instead research has shifted focus to designing new loss functions. Recent work has shown that while PLM-based models produce strong results, systems trained solely with maximum likelihood estimation (MLE) are poor at telling *good* from *bad* summaries (Sun and Li, 2021; Liu et al., 2022). An example of this is depicted in Table 2.6:

Sampled Summary	Rank f_θ	Rank $\mathcal{M}$	$\mathcal{M}$
An ongoing heatwave in West Australia could see [...]	1	5	57.3
A heatwave in West Australia could see temperature [...]	2	4	57.7
Australia could face some of its hottest temperature [...]	3	2	63.4
The Bureau of Meteorology warns that the towns of [...]	4	3	62.9
Two West Australian towns broke their records [...]	5	1	72.5

Table 2.6: Five sampled summaries from an abstractive system (Lewis et al., 2020). The first column contains a prefix of the sampled summary, the second column shows its rank wrt. to the summarizer, and the third column its rank wrt. $\mathcal{M}$.

In Table 2.6 it is clear that the generated summaries are negatively correlated with the evaluation metric (Pearson's $\rho = -0.9$), showing that the model is very poor at estimating (relative) summary quality. This is further emphasized by the large score difference between the lowest and highest scoring summary of 15 absolute points.

This problem was first described by Sun and Li (2021). As it turns out, abstractive systems trained purely with MLE are only slightly better than chance at ranking the worst from the best of its own sampled summaries (Liu et al., 2022). To address this Sun and Li (2021) propose an auxiliary *contrastive loss* term to encourage the model to produce scores that align with $\mathcal{M}$. This motivates a model to produce scores that reflect a summary's *quality* and thus enforces a strict *order* between summaries of different quality. Since the model is a generative sequence-to-sequence model, a summary is assigned

a score by computing its mean log probability as according to the model, p_θ:

$$f_\theta(S) = \frac{\sum \log p_\theta(S_i)}{|S|} \tag{2.22}$$

Using this scoring function, a model can be *calibrated* by penalizing it if a generated summary, S, scores higher than the reference summary R:

$$\mathcal{L}_{\text{aux}} = \max(0, f_\theta(R) - f_\theta(S) + \gamma) \tag{2.23}$$

Here γ is a fixed margin that enforces a minimum distance between S and R. Applying this loss showed to produce improvements across several benchmark datasets for summarization, showing that adding a calibration loss term positively boosts the performance of a system.

Liu and Liu (2021) continued this line of research and extended the auxiliary loss function to enforce a strict order between *multiple sampled summaries*. Instead of using the same model for generating and ranking, they opted to train a separate model to rank the summaries generated by the abstractive system.

$$\begin{aligned}\mathcal{L} = &\sum_i \max(0, f_\theta(R) - f_\theta(C_i) + \gamma) \\ &+ \sum_i \sum_{j>i} \max(0, f_\theta(C_j) - f_\theta(C_i) + \gamma)\end{aligned} \tag{2.24}$$

Here C denotes the sampled summaries and are sorted in descending order according to $\mathcal{M}$, such that $\mathcal{M}(c_i, R) > \mathcal{M}(c_{i+1}, R)$. γ again denotes a margin between summary scores. This is identical to the standard practice in machine translation (Shen et al., 2004) and turns abstractive summarization into a two-step process:

1. Sample candidate summaries from abstractive summarizer.

2. Rank candidates and return the highest scoring summary.

A central addition presented in this work is that the ranking model is exposed to a diverse set of summaries. This is achieved by utilizing *diverse beam search* (Vijayakumar et al., 2018) which can be applied to any sequence-to-sequence model. This avoids sampling similar summaries that differ only slightly from each other which results in a weak ranking model[9].

[9]Beam search explores the search space in a greedy left-right fashion retaining only the top k sequences that are often very similar.

Most recently, Liu et al. (2022) refined the efforts of the previous two approaches. The paper describes fine-tuning an abstractive summarizer in the same way as Sun and Li (2021) but using the extended contrastive loss in Equation 2.24 as proposed by Liu and Liu (2021). This resulted in a single summarizer with capabilities in both *generating text* and *scoring text*.

2.5.3 Cross-lingual Summarization

The information overload faced by society is not limited to the English-speaking community. According to w3techs.com[10] only 25.9% of internet users are proficient in English. Meanwhile, 72% of web users prefer consuming information in their native language (Kelly, 2012). To accommodate the need to consume information in a user's desired language there is a growing body of research on developing tools that enable knowledge dissemination across language barriers. This line of work is called *cross-lingual NLP*, with its efforts in summarization called *cross-lingual summarization* (CLS). While machine translation (MT) has come a long way, there are benefits to directly consuming relevant information conveyed in a different language instead of having to skim translated documents for potentially important information.

The goal of CLS is to produce a summary in a language that differs from the language of the source document. This implies that for the model to be a CLS summarizer it must take an input document that is written in a language that differs from the summary language, otherwise, it conflates to monolingual summarization.

Pipeline Systems

The first CLS systems were pipeline systems. A pipeline system is a combination of multiple single-purpose NLP systems that are merged into a single system by feeding the output of each model to the subsequent model. For CLS this means combining a machine translation model and a monolingual summarizer. Such a model allows two configurations which are defined by the order of execution, by either translating or summarizing first. Specifically, a pipeline system can be configured as *translate-then-summarize* (TS) or summarize-then-translate (ST). Each configuration imposes different data requirements. TS assumes access to a translation system for the source language, while ST requires a monolingual summarizer for the source language. With the prevalent scarcity of monolingual summarizers and the widespread

[10] `https://w3techs.com/technologies/overview/content_language`

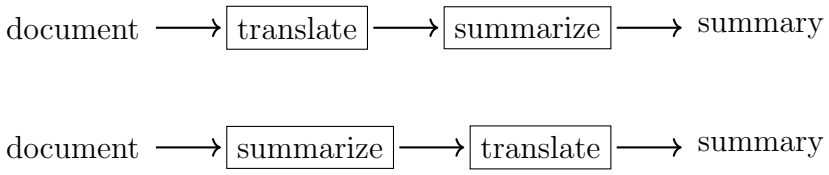


Figure 2.9: Configurations of a pipeline cross-lingual summarization system

availability of translation models, there are obvious benefits to considering TS when applying CLS from low-to-high-resource language pairs, and likewise, TS when the source language is a high-resource. With summarization being the less mature technology of the two, this rule makes the most use of available monolingual summarization data.

Orăsan and Chiorean (2008) were among the first to develop a CLS system. They developed an ST system for multi-document summarization of Romanian news articles into English. They employed an extractive Romanian summarizer and machine-translated the summaries into English. They found that poor translations of perhaps reasonable Romanian summaries did not produce useful English summaries due to errors in both summaries and translations, propagating errors. Wan et al. (2010) proposed a TS system from English to Chinese that machine-translated the English source document into Chinese and summarized the document with an extractive Chinese summarizer. To avoid including poor translations in the summary, translation scores were introduced as features of the summarizer. Wan et al. (2019) also developed a TS system from English to Chinese and similarly incorporated translation scores. Instead of using translation scores of sentences, the quality of multiple full-length summaries where considered.

End-to-End Systems

Much like abstractive systems so has research in CLS increasingly favored sequence-to-sequence designs resulting in pipeline evidence being rare. Sequence -to-sequence models provide an elegant end-to-end solution that removes the need for explicit translation and summarization steps, producing a single system that supports the direct summarization of documents in different languages. This may circumvent cascading errors but relies on large amounts of training data. As a result, most CLS efforts focus on exploring methods for obtaining or constructing CLS data.

Ouyang et al. (2019) investigated CLS for four languages, summarizing Somali, Swahili, Tagalog, and Arabic documents into English. With the

lack of a real CLS dataset, the English NYT (Sandhaus, 2008) was *back-translated* (Sennrich et al., 2016b) to obtain cross-lingual data and used this to train a pointer-generator system (See et al., 2017; Vinyals et al., 2015) for each language pair. They found that end-to-end CLS was feasible and model performance benefited from training on a mixture of languages, making the system more robust and benefiting from the transferability of related languages.

Cao et al. (2020) developed a Chinese-English CLS system by training a system on back-translated data and adding a custom back-translation loss that encourages the model to align text representations between languages. They report improvements over pipeline-based methods but leave out implementation details of the summarizer.

Ladhak et al. (2020) developed a cross-lingual dataset derived from the wiki-site *WikiHow* and fine-tune a multilingual PLM on it. The results show that even with access to large amounts of cross-lingual data, an end-to-end system does not immediately outperform a pipeline approach. Rather, the results suggest that end-to-end can be made to perform on par with pipeline methods but require back-translated samples to close the gap.

Summary

This chapter provides a brief summary of automatic text summarization. It introduces the notion of a summary and the various summary types and formalized a generalized goal of the task. Evaluation metrics, including n-gram and embedding-based metrics, are briefly introduced. Neural networks for text summarization are also introduced, showing that models are categorized into either *models that score text* or *models that generate text*. The chapter describes common datasets that are necessary to build summarizers, introduce the extractive, abstractive, and cross-lingual paradigms as well as introduce existing state-of-the-art systems.

This concludes what represents the background section of this thesis. The next two chapters will shift to outlining the scientific contributions included in this thesis, first within resource creation (Chapter 3) and then simplification of summarization systems (Chapter 4).

Chapter 3

Resource Creation

The previous chapter provided a background discussion and introduced some of the fundamental components of text summarization. This chapter now shifts to describing the contributions made by this thesis which address a central challenge faced by summarization technology, namely, *data*. The severe lack of (summarization) data for languages other than English prevents most languages from developing summarizers as they lack a minimal data foundation. This chapter outlines the efforts to close the apparent resource gap and introduces three large-scale datasets. The contributions made by this thesis to resource creation are:

1. The **first Danish text corpora**. A high-quality and diverse data foundation that enables investigating data-intensive models for Danish which has become a necessity for state-of-the-art language technology.

2. The **first Danish summarization dataset**. A resource that enables summarization development and research for the Danish language. By being *large-scale*, it enables research that requires large amounts of data and improves the multilingual data landscape for text summarization.

3. The **largest and most diverse multilingual text summarization dataset** covering 92 languages, across 35 writing scripts, further strengthening multilingual text summarization.

4. A language agnostic data collection method and the first investigation of the **efficacy of automatic methods for multilingual dataset creation**, providing opportunities and limitations to existing methods.

3.1 Creating a Danish Text Corpus

Leon Strømberg-Derczynski, Manuel Ciosici, Rebekah Baglini, Morten H. Christiansen, Jacob Aarup Dalsgaard, Riccardo Fusaroli, Peter Juel Henrichsen, Rasmus Hvingelby, Andreas Kirkedal, Alex Speed Kjeldsen, Claus Ladefoged, Finn Årup Nielsen, Jens Madsen, Malte Lau Petersen, Jonathan Hvithamar Rystrøm, and Daniel Varab. 2021. **The Danish Gigaword Corpus**. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 413-421, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Access to large amounts of task-specific *labeled data* is pivotal to the performance of modern data-driven NLP systems. However, in the age of pretraining perhaps equally important is access to large amounts of *unlabeled text data*. With contemporary summarization system designs being de facto developed on top of PLMs, current designs assume access to a PLM. With abundant text data being predominantly an artifact of a few privileged languages, the remainder of languages faces a challenging reality. Specifically, anyone that wishes to develop a summarizer that targets other languages than those with an already established PLM can not, and is prevented from benefiting from incremental progress that assumes access to a PLM. This challenge is widely acknowledged for the Danish language (Kirkedal et al., 2019; Kirchmeier et al., 2019), and there is a dire need to ensure access to linguistic resources to ensure future progress on Danish NLP.

The Danish Gigaword Corpus addresses this lack of text data for the Danish language by introducing a high-quality curated text corpus containing more than 1 billion words. It includes of a wide range of text domains, including legal documents, social media text, dialogue, and literature, providing a diverse data foundation for future data-driven Danish NLP research. Although not large enough to match the size of datasets currently used to train PLMs, research has shown that even relatively small amounts of data allow PLMs to adapt to new languages and domains (Rosset, 2019; Gao et al., 2020; Floridi and Chiriatti, 2020). In the dataset's short lifespan, it has already been embraced by researchers and practitioners and has enabled two Danish PLMs (Højmark-Bertelsen, 2021; Ciosici and Derczynski, 2022).

3.2 Creating a Danish Summarization Dataset

Daniel Varab and Natalie Schluter. 2020. **DaNewsroom: A Large-scale Danish Summarisation Dataset**. *In Proceedings of the 12. Language Resources and Evaluation Conference*, pages 6731-6739, Marseille, France. European Language Resources Association.

Summarization data is a necessity to train any competitive summarizer, let alone to evaluate the performance of current and future systems. The *DaNewsroom* dataset addresses the complete lack of summarization data for the Danish language and introduces the first Danish summarization dataset ever. With more than 1.1 million document-summary pairs it is comparable to, or larger, than the English benchmark datasets commonly included in summarization literature. To build the dataset a technique for building summarization datasets from news sites is developed and extensively documented, thus, contributing a scalable language-agnostic method to collect large amounts of summarization data for theoretically any language that has online news outlets. Requiring only a list of URLs the method uses a few heuristics that encourage data collection of high-quality summarization data. The contribution of this paper is, twofold, a large summarization dataset and a language-agnostic data collection method that enables the future creation of summarization datasets for *any language*.

The paper can also be seen as a reproduction effort of previous summarization dataset initiatives for summarization (Grusky et al., 2018; Nguyen and Daumé III, 2019). The DaNewsroom dataset explores an existing method in a new setting by applying it simultaneously to a *a non-English language as well as collecting from multiple websites*. My work draws a similar conclusion in that summarization datasets can with relatively small efforts be produced even for a relatively small language, paving the way for future expansions to other languages, and showing promising directions in multilingual summarization.

Most importantly, it fills an immediate resource gap for the Danish language. The dataset has sparked immediate interest and has been the subject of interest from over 20 industry practitioners, students, and fellow researchers. The DaNewsroom dataset has laid the ground for the research efforts of several university students that have shown interest in developing summarizers as part of their studies, emphasizing the dataset's value as an enabler for summarization technology for the Danish language (Hansen et al., 2022; Nielsen and Veile, 2020).

3.3 Creating Summarization Datasets for 92 Languages

Daniel Varab and Natalie Schluter. 2021. **MassiveSumm: a very large-scale, very multilingual, news summarisation dataset**. *In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10150-10161, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

With the success of creating a Danish summarization dataset a natural question arises: *Can this method be used to create summarization datasets for all languages?* To explore this, and to address the lack of summarization datasets for a wide set of languages, the *MassiveSumm* dataset was created. Since the method developed for the Danish summarization dataset was *language agnostic*, it is possible to apply it to any language. This also allows exploring the efficacy of the automatic method when applied to a more (linguistically) diverse set of languages. The study documents the process of building a multilingual summarization dataset, larger in scope and with a more diverse set of languages than any previous effort. The dataset, *MassiveSumm*, contains 12.3 million document-summary pairs across 92 languages, 38 language families, and 35 writing scripts. As a side-effect of some data not having suitable summaries, the dataset also includes 61.5 GB of raw text data across all languages. Contrary to past (and concurrent) evidence (Scialom et al., 2020; Hasan et al., 2021) the resulting dataset shows that the method generalizes poorly to the majority of languages, and only works well on a particular subset of languages. Specifically, the method produces much more data for Indo-European languages, with these languages constituting almost the entire dataset (73%). The study shows that existing heuristics-based data collection methods *do not apply, nor scale to arbitrary languages*. Specifically, for many languages, these approaches are partially or completely ineffective or produce very little data. It concludes that currently only relying on web content for data (labeled or not) heavily favors Indo-European languages, thus, calling for alternative methods for low-resource non-Indo-European languages. This finding supplements concurrent related work that has emphasized the limitations of automatically derived web-based datasets (Luccioni and Viviano, 2021; Kreutzer et al., 2022; Abadji et al., 2022; Jansen et al., 2022)

Chapter 4

Simplified Systems

This chapter sheds light on a different matter of summarization and represents the second round of contributions of this thesis: *system designs*. Summarization systems have developed many improvements over the years. To further improve summarization systems and push the frontiers of text summarization, system designs are increasingly more specialized and are becoming more complex. While complexity can provide performance benefits, it often comes at a cost of pragmatism, making systems both hard to maintain and costly to run. While specialized solutions are warranted it is important to recognize that there are benefits to prioritizing simple systems and standardized designs. This chapter outlines two studies that show that simple, reliable and pragmatic systems can be just as competitive as complex custom system architectures. The contributions are:

1. A **novel summarization paradigm** that allows a **unified system to support multiple summary types**, generating both extractive and abstractive summaries on demand.

2. A novel **extractive inference algorithm for sequence-to-sequence models** and evidence that indicates that current abstractive systems can produce extractive summaries that are on par with state-of-the-art extractive systems.

3. A timely **reevaluation of end-to-end designs for cross-lingual summarization** on 39 languages, showing that contrary to recent research trends, pipeline designs often produce superior results.

4.1 Generative Extractive Summarization

> Daniel Varab and Yumo Xu. 2023. **Abstractive Summarizers are Excellent Extractive Summarizers**. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 330–339, Toronto, Canada. Association for Computational Linguistics.

Model designs for extractive and abstractive summarization systems have in recent years diverged to the extent that each paradigm is developed in isolation. While this has led to some improvements to each summary type, it is making systems from each paradigm increasingly incompatible, preventing immediate synergies between them. Meanwhile, it has been shown that combining the paradigms can lead to both improved content selection (Kedzie et al., 2018; Gehrmann et al., 2018) and more control over generation summaries (Dou et al., 2021). Such efforts currently resort to multiple systems trained on different data, pipelining predictions at inference time.

The paper *Abtractive Systems are Excellent Extractive Summarizers* conceptualizes a novel paradigm along an inference algorithm that allows abstractive summarizers to produce both abstractive and extractive summaries. Taking advantage of recent advancements in abstractive summarization which have enabled summarizers to *estimating summary quality*, the paper suggests capitalizing on this newfound property to use abstractive systems to score sentences and produce extractive summaries.

This, for the first time, shows that a single system can model extractive and abstractive summarization simultaneously, and fundamentally challenges the need to develop separate models for each summary-types. It goes to show that abstractive systems can without difficulty be extended to support other types than abstractive summaries and suggests that this might be through new inference algorithms.

Unifying extractive and abstractive summarization into a single design is useful. It removes the need for practitioners to build and maintain separate systems for each summary type and shift the focus on building a single versatile dual-purpose system. This also falls in line with the efforts that frame all tasks as text generation, through transfer-learning, prompting, or zero-shot inference of large PLMs like T5 (Raffel et al., 2020), PaLM (Chowdhery et al., 2022), GPT-3 (Floridi and Chiriatti, 2020), and the recent excitement surrounding ChatGPT.

4.2 Robust Cross-lingual Summarization

Daniel Varab and Christian Hardmeier. **With Good MT There is No Need For End-to-End: A Case for Translate-then-Summarize Cross-lingual Summarization**. *Conference paper under review.*

Cross-lingual summarization (CLS) expands the notion of summarizing information within one language (monolingual) to summarizing information across language barriers (cross-lingual) by conveying information expressed in a *source language* into another desired *target language*. A straightforward approach to modeling CLS is to combine a machine translation system and a monolingual summarizer into a single *pipelined system*. With the successes of generative neural networks, a single model can be trained in an *end-to-end* fashion with access to cross-lingual summarization data using a sequence-to-sequence model. This approach has in recent years gained sizable traction, leaving the impression that end-to-end designs are a viable design choice for CLS. However, a closer look reveals that conclusions are often based on experiments with language for which there is plenty of CLS data available, or make comparisons to underpowered or even undocumented pipeline baselines.

The paper *With Good MT There is No Need For End-to-End: An Empirical Study of Cross-lingual Summarization* conducts a simple but timely comparative study to uncover the efficacy of end-to-end designs and addresses a potentially misleading emerging best practice. The study compares proposed end-to-end designs with accessible but strong pipeline designs and finds no evidence that end-to-end designs should be preferred over traditional pipeline designs. Rather, end-to-end systems *can perform CLS* but rarely outperform pipeline baselines.

This paper contributes with a more nuanced view of the state of CLS by showing that end-to-end for CLS is not *not yet* fit for practical use and that it remains a topic reserved for academic inquiry at the present moment.

Chapter 5

Conclusions

5.1 Main Conclusions

In this final chapter, each contribution made by this thesis is concluded with a closing remark. First, the method used to create the included datasets is discussed, then the opportunities of a unified summarization system are reflected upon, and lastly, the efficacy of end-to-end cross-lingual summarization is discussed.

Expanding Summarization to New Languages

This thesis has contributed with an effective method to automatically construct summarization datasets and has shown that applying this method successfully creates large-scale summarization datasets, even for low-resource languages. Successfully creating two large-scale datasets for a low-resource language like Danish using methods developed for English is, however, not necessarily representable for other low-resource languages. Despite being a relatively small language, the Danish language has been subject to aggressive digitalization resulting in a relatively large presence on the internet (Schou and Hjelholt, 2019). This most likely contributes to the dataset being of equal size to English datasets. This also means that although the method has been successful for *one low-resource language*, we must be careful not to assert that the method scales well for *any low-resource language*. This was demonstrated with the MassiveSumm dataset, showing that current methods *do not* transfer to arbitrary languages. Consider Thai, a language with a similar online presence, but spoken by nearly 11 times more speakers than Danish. In the final version of MassiveSumm, Thai represented less than $\frac{1}{10}$th of that collected Danish data for the DaNewsroom dataset. This can poten-

tially be attributed to the efficacy of the method, but may also be influenced by annotator bias. While the methodology is largely automatic, it requires a manually created list of news sites to collect data. With the lists of news sites being created mostly by curated by non-speakers, MassiveSumm is likely to be suboptimal by not considering resourceful news sites.

Unifying abstractive and extractive designs

The concepts of abstractive and extractive summaries are well-established and have been described thoroughly already in early summarization literature. The modeling of these concepts, on the other hand, has throughout time been driven primarily by technical limitations. With fluent text generation techniques only recently becoming accessible, extractive summarizers have evolved as a *necessary simplification*. Recent developments have made it easy to build systems that generate text, causing a surge of abstractive designs that capitalize on being *not extractive*. This disconnect prevents an immediate lost opportunity as the extractive and abstractive pose convenient complimentary properties. Summarizing with extracts resolves almost every challenge faced by abstracts and vice versa. For example, abstractive systems are widely criticized for their tendency to fabricate information not present in the source document. Meanwhile, extractive systems are virtually incapable of fabricating information. In the opposite direction, extractive systems are rigid and often unable to match user needs. Summarizers that embrace both summary types are not only practical but powerful and sets the stage for a new research direction on hybrid models and inference algorithms.

End-to-End Cross-lingual Summarization Research

Current evidence in favor of end-to-end designs for cross-lingual summarization approaches is brittle and provides little evidence that such designs should be preferred over traditional pipeline methods. The argument in favor of end-to-end systems over pipelined systems is *performance* as this provides a solution to circumvent cascading error effects. For it to be an appropriate solution, a well-designed system should perform on par with or better than a *strong pipeline baseline*. Current evidence suggests this is possible but assumes access to large amounts of cross-lingual summarization data and pre-trained language models.

This touches upon a central issue faced by end-to-end designs, which makes them unlikely to transfer to most practical settings. Monolingual summarization data can be collected from the web but cross-lingual data is a

much less common type of data. This suggests that even if current literature suggests that end-to-end systems are capable of outperforming pipeline baselines, they do so under the assumption of access to data that is rare or does not exist.

Meanwhile, the individual progress of machine translation and monolingual summarization continues to advance. Both tasks are undergoing rapid progress whilst building on a firm foundation of data, evaluation methods and analysis. Since end-to-end approaches have yet shown easily better pipeline methods, it begs the question: Will they ever? This thesis does not challenge the *feasibility* of end-to-end modeling but rather challenges the *efficacy* of the approach. While this does not rule out future innovations, it appears that a pipeline approach for cross-lingual summarization is a predictable, reliable and transparent choice at present.

5.2 Future Work

Obtaining the necessary data to develop modern summarization from the web is an effective strategy, but as covered in the previous sections, only under certain constraints. While especially low-resource languages are unlikely to equally benefit from the method, it does provide a means to obtain *large quantities* of summarization data for many languages. However, an immediate weakness of the method is that it does not provide a means of obtaining *quality* summarization data. While the method does include some heuristics to *encourage* including good summaries, these heuristics do not guarantee that summaries are useful. Unfortunately, quality estimating a summary is no easy task and it is closely related to actually producing a summary. To improve the quality of automatically collected datasets, future work should focus on developing techniques that allow quality estimating summarization data to produce higher-quality summarization datasets.

Performing competitive text summarization with simple and pragmatic summarizers is completely feasible without specialized models. This thesis emphasized this in two research efforts, one conceptualizing that multiple summary types can be supported by a single model, and one highlighting the benefits of the effectiveness of (to some dated) pipeline designs. The former effort is the first to date to explore unified designs to support multiple summary types and paves the way for future work on unified systems. The work included in this thesis shows that abstractive systems can perform competitive extractive summarization. An immediate next step is to explore the

limitations of a unified approach, exploring how to further improve extractive summarization.

Cross-lingual summarization research is still in its infancy. The contributions of this thesis focused on highlighting that end-to-end systems might not necessarily be the default design choice as they by posing unrealistic data requirements and not convincingly outperforming pipeline methods. Future work would benefit from an exhaustive comparison of the two paradigms, investigating behavior in different data settings and recommendations for language pairs.

As the abundance of available information grows, text summarization becomes an increasingly important tool in today's society. With the rate of information being created every day, it is overwhelming for people to keep up. Text summarization offers an elegant solution to this problem by providing people with a more palatable means of digesting large amounts of information. Moreover, summarization can also assist in enhancing the accessibility of information, allowing people with reading difficulties or limited literacy levels to understand the main points of documents. By improving summarization in this thesis, summarization is one step closer to democratizing access to information by empowering individuals with knowledge.

Bibliography

Hans Peter Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165, 1958.

Inderjeet Mani and Mark T. Maybury. Automatic summarization. *Computational Linguistics*, 28:221–223, 2002.

Harold P Edmundson. New methods in automatic extracting. *Journal of the ACM (JACM)*, 16(2):264–285, 1969.

Harold Borko and Charles L. Bernier. Abstracting concepts and methods. *Academic Press*, 1975.

Inderjeet Mani, Gary Klein, David House, Lynette Hirschman, Therese Firmin, and Beth Sundheim. Summac: a text summarization evaluation. *Natural Language Engineering*, 8(1):43–68, 2002.

Ani Nenkova, Kathleen McKeown, et al. Automatic summarization. *Foundations and Trends in Information Retrieval*, 5(2–3):103–233, 2011.

E. Jeffrey Conklin and David D. McDonald. Salience: The key to the selection problem in natural language generation. In *20th Annual Meeting of the Association for Computational Linguistics*, pages 129–135, Toronto, Ontario, Canada, June 1982. Association for Computational Linguistics.

Maxime Peyrard. A simple theoretical model of importance for summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1059–1073, Florence, Italy, July 2019. Association for Computational Linguistics.

Antoine Bosselut, Esin Durmus, Varun Prashant Gangal, Sebastian Gehrmann, Yacine Jernite, Laura Perez-Beltrachini, Samira Shaikh, and

Wei Xu, editors. *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, Online, August 2021. Association for Computational Linguistics.

Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409, 2021.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022.

Sumit Chopra, Michael Auli, and Alexander M. Rush. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98, San Diego, California, June 2016. Association for Computational Linguistics.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany, August 2016. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

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, June 2019. Association for Computational Linguistics.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. 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 563–578, Hong Kong, China, November 2019. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019a.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Yann Le Cun and Yoshua Bengio. Word-level training of a handwritten word recognizer based on convolutional neural networks. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition, vol. 3-Conference C: Signal Processing (Cat. No. 94CH3440-5)*, volume 2, pages 88–92. IEEE, 1994.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. Improving named entity recognition by external context retrieving and cooperative learning. 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 1800–1812, Online, August 2021a. Association for Computational Linguistics.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. Automated concatenation of embeddings for struc-

tured prediction. 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 2643–2660, Online, August 2021b. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. 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 3730–3740, Hong Kong, China, November 2019. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics.

Kai Sun, Dian Yu, Dong Yu, and Claire Cardie. Improving machine reading comprehension with general reading strategies. 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 2633–2643, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

Gladys Tyen, Mark Brenchley, Andrew Caines, and Paula Buttery. Towards an open-domain chatbot for language practice. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 234–249, Seattle, Washington, July 2022. Association for Computational Linguistics.

John Firth. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*, pages 10–32, 1957.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf.

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 2020. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020.

Yukun Zhu, Ryan Kiros, Rich 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 *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures. In Piotr Bański, Adrien Barbaresi, Hanno Biber, Evelyn Breiteneder, Simon Clematide, Marc Kupietz, Harald Lüngen, and Caroline Iliadi, editors, *Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019*, pages 9 – 16, Mannheim, 2019. Leibniz-Institut für Deutsche Sprache. doi: 10.14618/ids-pub-9021. URL http://nbn-resolving.de/urn:nbn:de:bsz:mh39-90215.

Julien Abadji, Pedro Ortiz Suarez, Laurent Romary, and Benoît Sagot. Towards a Cleaner Document-Oriented Multilingual Crawled Corpus. *arXiv e-prints*, art. arXiv:2201.06642, January 2022.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe,

Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmungkol Sarin, Sokhar Samb, Benoît Sagot, Clara Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Ortiz Suarez, Iroro Orife, Kelechi Ogueji, Andre Niyongabo Rubungo, Toan Q. Nguyen, Mathias Müller, André Müller, Shamsuddeen Hassan Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhalov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, Mathias Jenny, Orhan Firat, Bonaventure F. P. Dossou, Sakhile Dlamini, Nisansa de Silva, Sakine Çabuk Ballı, Stella Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, Pallavi Baljekar, Israel Abebe Azime, Ayodele Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta Agrawal, and Mofetoluwa Adeyemi. Quality at a Glance: An Audit of Web-Crawled Multilingual Datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72, 01 2022. ISSN 2307-387X. doi: 10.1162/tacl_a_00447. URL `https://doi.org/10.1162/tacl_a_00447`.

René Haas and Leon Derczynski. Discriminating between similar nordic languages. In *Proceedings of the Eighth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 67–75, Kiyv, Ukraine, April 2021. Association for Computational Linguistics.

Khanh Nguyen and Hal Daumé III. Global Voices: Crossing borders in automatic news summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 90–97, Hong Kong, China, November 2019. Association for Computational Linguistics.

Alex Wang, Richard Yuanzhe Pang, Angelica Chen, Jason Phang, and Samuel R. Bowman. SQuALITY: Building a long-document summarization dataset the hard way. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1139–1156, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL `https://aclanthology.org/2022.emnlp-main.75`.

Emi Ishita, Satoshi Fukuda, Yoichi Tomiura, and Douglas W. Oard. Using text classification to improve annotation quality by improving anno-

tator consistency. *Proceedings of the Association for Information Science and Technology*, 57(1):e301, 2020. doi: https://doi.org/10.1002/pra2.301. URL `https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/pra2.301`.

Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium, October-November 2018a. Association for Computational Linguistics.

Max Grusky, Mor Naaman, and Yoav Artzi. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. 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 708–719, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34, 2003.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28, 2015.

Meng Cao, Yue Dong, and Jackie Cheung. Hallucinated but factual! inspecting the factuality of hallucinations in abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3340–3354, Dublin, Ireland, May 2022. Association for Computational Linguistics.

Baotian Hu, Qingcai Chen, and Fangze Zhu. LCSTS: A large scale Chinese short text summarization dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1967–1972, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

Faisal Ladhak, Esin Durmus, Claire Cardie, and Kathleen McKeown. WikiLingua: A new benchmark dataset for cross-lingual abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4034–4048, Online, November 2020. Association for Computational Linguistics.

Fajri Koto, Jey Han Lau, and Timothy Baldwin. Liputan6: A large-scale Indonesian dataset for text summarization. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 598–608, Suzhou, China, December 2020. Association for Computational Linguistics.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. MLSUM: The multilingual summarization corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8051–8067, Online, November 2020. Association for Computational Linguistics.

Yumo Xu and Mirella Lapata. Text summarization with oracle expectation. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=HehQobsr0S`.

Rada Mihalcea and Paul Tarau. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain, July 2004. Association for Computational Linguistics.

Günes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479, 2004.

Daniel Gillick, Benoit Favre, and Dilek Z. Hakkani-Tür. The icsi summarization system at tac 2008. *Theory and Applications of Categories*, 2008.

Jianpeng Cheng and Mirella Lapata. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494, Berlin, Germany, August 2016. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Ranking sentences for extractive summarization with reinforcement learning. 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 1747–1759, New Orleans, Louisiana, June 2018b. Association for Computational Linguistics.

Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–663, Melbourne, Australia, July 2018. Association for Computational Linguistics.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online, July 2020. Association for Computational Linguistics.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6, 1993.

Shashi Narayan, Joshua Maynez, Jakub Adamek, Daniele Pighin, Blaz Bratanic, and Ryan McDonald. Stepwise extractive summarization and planning with structured transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4143–4159, Online, November 2020. Association for Computational Linguistics.

Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. ETC: Encoding long and structured inputs in transformers. In

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 268–284, Online, November 2020. Association for Computational Linguistics.

Xingxing Zhang, Furu Wei, and Ming Zhou. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5059–5069, Florence, Italy, July 2019b. Association for Computational Linguistics.

Yicheng Zou, Lujun Zhao, Yangyang Kang, Jun Lin, Minlong Peng, Zhuoren Jiang, Changlong Sun, Qi Zhang, Xuanjing Huang, and Xiaozhong Liu. Topic-oriented spoken dialogue summarization for customer service with saliency-aware topic modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 14665–14673, 2021.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016a. Association for Computational Linguistics.

Nitish Shirish Keskar, Bryan McCann, Lav Varshney, Caiming Xiong, and Richard Socher. CTRL - A Conditional Transformer Language Model for Controllable Generation. *arXiv preprint arXiv:1909.05858*, 2019.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

Shichao Sun and Wenjie Li. Alleviating exposure bias via contrastive learning for abstractive text summarization. *arXiv preprint arXiv:2108.11846*, 2021.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. BRIO: Bringing order to abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2890–2903, Dublin, Ireland, May 2022. Association for Computational Linguistics.

Yixin Liu and Pengfei Liu. SimCLS: A simple framework for contrastive learning of abstractive summarization. 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 1065–1072, Online, August 2021. Association for Computational Linguistics.

Libin Shen, Anoop Sarkar, and Franz Josef Och. Discriminative reranking for machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 177–184, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics.

Ashwin Vijayakumar, Michael Cogswell, Ramprasaath Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse beam search for improved description of complex scenes. *Proceedings of the 32th AAAI Conference on Artificial Intelligence*, 2018.

Nataly Kelly. Speak to Global Customers in Their Own Language. `https://hbr.org/2012/08/speak-to-global-customers-in-t`, 2012. Accessed: 2023-15-02.

Constantin Orăsan and Oana Andreea Chiorean. Evaluation of a cross-lingual Romanian-English multi-document summariser. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA).

Xiaojun Wan, Huiying Li, and Jianguo Xiao. Cross-language document summarization based on machine translation quality prediction. In *Proceedings*

of the 48th Annual Meeting of the Association for Computational Linguistics, pages 917–926, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

Xiaojun Wan, Fuli Luo, Xue Sun, Songfang Huang, and Jin-ge Yao. Cross-language document summarization via extraction and ranking of multiple summaries. *Knowledge and Information Systems*, 58(2):481–499, 2019.

Jessica Ouyang, Boya Song, and Kathy McKeown. A robust abstractive system for cross-lingual summarization. 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 2025–2031, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

Evan Sandhaus. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752, 2008.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany, August 2016b. Association for Computational Linguistics.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. *Advances in neural information processing systems*, 28, 2015.

Yue Cao, Hui Liu, and Xiaojun Wan. Jointly learning to align and summarize for neural cross-lingual summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6220–6231, Online, July 2020. Association for Computational Linguistics.

Andreas Kirkedal, Barbara Plank, Leon Derczynski, and Natalie Schluter. The lacunae of Danish natural language processing. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 356–362, Turku, Finland, September–October 2019. Linköping University Electronic Press.

Sabine Kirchmeier, Peter Juel Henrichsen Philip Diderichsen, and Nanna Bøgebjerg Hansen. Dansk sprogteknologi i verdensklasse. In *The Danish Language Council*, 2019.

Corby Rosset. Turing-nlg: A 17-billion-parameter language model. *Microsoft Blog*, 2019.

Luciano Floridi and Massimo Chiriatti. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30(4):681–694, 2020.

Malte Højmark-Bertelsen. Ælæctra - a step towards more efficient danish natural language processing, 2021. URL `https://github.com/MalteHB/-l-ctra/`.

Manuel R Ciosici and Leon Derczynski. Training a t5 using lab-sized resources. *arXiv preprint arXiv:2208.12097*, 2022.

Ida Bang Hansen, Sara Kolding, and Katrine Nymann. Automatic abstractive summarisation in danish, 2022. URL `https://github.com/idabh/data-science-exam`.

Lukas Christian Nielsen and Sebastian Lindegaard Veile. Automatic text summarization for danish using bert, 2020. URL `https://www.derczynski.com/itu/docs/danish-summarisation.pdf`.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703, Online, August 2021. Association for Computational Linguistics.

Alexandra Luccioni and Joseph Viviano. What's in the box? an analysis of undesirable content in the Common Crawl corpus. 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 182–189, Online, August 2021. Association for Computational Linguistics.

Julien Abadji, Pedro Ortiz Suarez, Laurent Romary, and Benoît Sagot. Towards a cleaner document-oriented multilingual crawled corpus. In *International Conference on Language Resources and Evaluation*, 2022.

Tim Jansen, Yangling Tong, Victoria Zevallos, and Pedro Ortiz Suarez. Perplexed by quality: A perplexity-based method for adult and harmful content detection in multilingual heterogeneous web data, 2022. URL `https://arxiv.org/abs/2212.10440`.

Chris Kedzie, Kathleen McKeown, and Hal Daumé III. Content selection in deep learning models of summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1818–1828, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. GSum: A general framework for guided neural abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4830–4842, Online, June 2021. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

Jannick Schou and Morten Hjelholt. Digitalizing the welfare state: citizenship discourses in danish digitalization strategies from 2002 to 2015. *Critical Policy Studies*, 13(1):3–22, 2019.

Paper I

The Danish Gigaword Corpus

Leon Derczynski
ITU Copenhagen
Denmark
ld@itu.dk

Manuel R. Ciosici
USC Information Sciences Institute
USA
manuelc@isi.edu

Rebekah Baglini
Aarhus University
Denmark

Morten H. Christiansen
Aarhus University & Cornell University
Denmark

Jacob Aarup Dalsgaard
Aarhus University
Denmark

Riccardo Fusaroli
Aarhus University
Denmark

Peter Juel Henrichsen
Danish Language Council
Denmark

Rasmus Hvingelby
Alexandra Institute
Denmark

Andreas Kirkedal
ITU Copenhagen
Denmark

Alex Speed Kjeldsen
University of Copenhagen
Denmark

Claus Ladefoged
TV2 Regionerne
Denmark

Finn Årup Nielsen
Technical University of Denmark
Denmark

Jens Madsen
Karnov Group
Denmark

Malte Lau Petersen
Aarhus University
Denmark

Jonathan Hvithamar Rystrøm
Aarhus University
Denmark

Daniel Varab
Novo Nordisk & ITU Copenhagen
Denmark

Abstract

Danish language technology has been hindered by a lack of broad-coverage corpora at the scale modern NLP prefers. This paper describes the Danish Gigaword Corpus, the result of a focused effort to provide a diverse and freely-available one billion word corpus of Danish text. The Danish Gigaword corpus covers a wide array of time periods, domains, speakers' socio-economic status, and Danish dialects.

1 Introduction

It is hard to develop good general-purpose language processing tools without a corpus that is broadly representative of the target language. Further, developing high-performance deep learning models requires hundreds of millions of tokens (Radford et al., 2019; Raffel et al., 2020). To address this gap for Danish, a North Germanic/Scandinavian language spoken primarily in Denmark, we propose an open giga-word corpus. This corpus is free to download and use, thus enabling researchers and organizations to further develop Danish NLP without worrying about licensing fees. The corpus is a first necessary step to allow Danish speakers to receive the many benefits of the powerful range of NLP technologies.

This paper details the Danish Gigaword Corpus (DAGW), a billion-word corpus of language across various dimensions, including modality, time, setting, and place.

It is tricky to collect such a corpus automatically: automatic language identification tools confound closely related languages, especially Danish and

Bokmål, and are likely to miss important data (Radford et al., 2019; Haas and Derczynski, 2021). Existing representations underperform for Danish: the multilingual FastText embeddings (Joulin et al., 2018) miss core Danish words such as "træls"; Multilingual BERT lacks sufficient support for the Danish vowel "å".[1]

To remedy this situation, we propose a Danish Gigaword Corpus. The overriding goals are to create a dataset that is (1) representative, (2) accessible, and (3) a general-purpose corpus for Danish.

2 Background

Today's NLP is generally data-intensive, meaning that large representative corpora tend to correlate with better models and better processing results. However, large representative corpora are available for only a small set of languages; there are fewer than ten manually-compiled gigaword-scale corpora, for example, and none for Danish.

Several substantial Danish text corpora have been compiled during recent decades. CLARIN-DK offers a variety of individual corpora of varying genres, annotations, and writing times. However, non-commercial licensing restricts corpus usage. Some major Danish corpora are related to dictionary production, as is the case for the 56 million words Korpus-DK available for search at the dictionary site ordnet.dk.[2] Leipzig Corpora Collection assembles Danish corpora from the Web, news sites, and Wikipedia (Goldhahn et al., 2012). The combined size of these corpora is orders of magnitude smaller than The Danish Gigaword Corpus. By themselves, these corpora do not meet the data size needs of modern language models.

Modern language models like T5 (Raffel et al., 2020) and GPT2 (Radford et al., 2019) are text-hungry, making automatic corpora construction attractive. Massive, monolithic, automatically collected datasets of web content, such as Common Crawl, support the training of large language models but suffer from quality issues (Radford et al., 2019) and bias (Ferrer et al., 2021). Models trained exclusively with such data quickly delve into generating toxic language (Gehman et al., 2020). Furthermore, the Danish section of Common Crawl is plagued by significant amounts of non-Danish content, in part due to the pervasive confusion between Danish and Norwegian Bokmål by highly multilingual language ID classifiers (Haas and Derczynski, 2021). Datasets derived exclusively from Common Crawl also have a bias toward webspeak and content from recent years, leaving models built over them sub-optimally prepared to process older Danish.

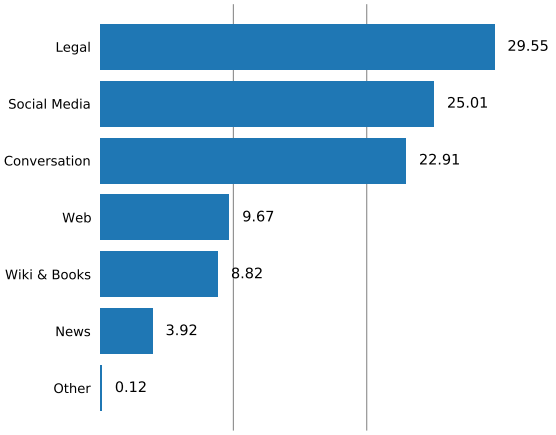


Figure 1: Content by domain (% of corpus).

The lack of a large and qualitative Danish corpus causes Danish NLP tools to lag behind equivalent tools for better-resourced languages, and the gap is increasing (Pedersen et al., 2012; Kirkedal et al., 2019; Kirchmeier et al., 2020).

The first gigaword corpus was the English Gigaword (Graff et al., 2003), consisting of roughly one billion (10^9) words of English-language newswire text. The content was single-genre, national and global newswire, published between 1994 and 2002. Other gigaword corpora emerged later, for French, Arabic, Chinese, and Spanish. Even Icelandic, a language with just over 360 000 speakers, has a healthy gigaword project (Steingrímsson et al., 2018).

3 Linguistic diversity

For a corpus to be useful for a wide range of applications, it must include a wide range of language, mixing domains, speakers, and styles (Biber, 1993). Failing to do this can lead to severe deficiencies in the data. For example, when NLP work started on social media text, the Wall Street Journal-trained part of speech taggers missed essential words such as "Internet" (due to the articles being from the late

[1] BotXO maintains a Danish BERT instance at `https://github.com/botxo/nordic_bert`. This model was trained exclusively on uncurated web text and, therefore, (a) has a spurious understanding of Danish among other languages and (b) is particularly susceptible to the kind of toxic language identified by Gehman et al. (2020).

[2] `http://ordnet.dk`

eighties and early nineties) and "bake", due to their domain.

Common Crawl's undirected collection of content often over-represents some dialects at the expense of other dialects. GeoWAC (Dunn and Adams, 2020) uses demographic information to construct English corpora that balance dialects. Unfortunately, a demographic- and Web-based approach underrepresents Danish dialects such as the endangered Bornholmsk dialect (Mortensen, 2016), which is almost absent from the Web.

These deficiencies do not form a solid basis for general-purpose NLP. So the Danish Gigaword Corpus captures and distributes as broad a range of Danish language use as possible, explicitly including language from a variety of settings (long-form writing, novels, social media, speeches, spontaneous speech), domains (news, politics, fiction, health, social media, law, finance), time periods (from the 1700s to present day), registers (formal, informal), and dialects (including, e.g., Bornholmsk and Sønderjysk).

4 Dataset construction

The Danish Gigaword Corpus consists of sections, with each section corresponding to a single source of text. Following prior efforts to construct broad-coverage datasets (Derczynski et al., 2016), sections are selected based on how well they help the corpus' coverage of Danish language use over a variety of dimensions, including: time of authorship; speech situation; modality; domain; register; age of utterer; dialect of utterer; socio-economic status of utterer. This is a strong, intentional departure from editions of English Gigaword that focused on newswire. Achieving some degree of representativeness (Biber, 1993) requires the inclusion of sources beyond newswire text. We provide an overview of The Danish Gigaword Corpus's content in Figure 1 and detail the sections in Table 1 and the appendix.

The Danish Gigaword Corpus follows the definition of genre used by Biber (1993), grounded in "situationally defined categories", such as a language style recognized by (or used to define) a community, such as news articles, personal letters, or online chat; a domain as a particular topical focus (or set of foci) that are discussed, such as biomedicine, politics, or gaming; and a medium as the means by which communication is conducted, such as writing, online chat, conversation, and so on. There is a natural overlap between medium and speech situations, but the delineation is beyond this work's scope.

While the goal of DAGW is to cover a range of genres, domains, and media, it is difficult to measure the prevalence of each of these across all Danish users, let alone then gather and redistribute this data. Therefore, the goal is to cover something of everything that can be feasibly included, without letting any particularly monolithic combination dominate (in contrast to, e.g., the 100% written newswire content of English Gigaword v1 or the 100% Common Crawl content of GeoWAC). Not every intersection between genres, domains, and media can be covered, nor represented proportionally, in the first version of this corpus. Table 1 contains an overview of the genres, domains, and modalities included in the Danish Gigaword Corpus.

4.1 Data and metadata unification

Each section is contained in one directory, named after the "prefix" for the section. Each file in a section represents a single UTF encoded document. Each section contains at least two functional files: one describing how the section is licensed and one describing metadata about each document. For multi-speaker corpus sections, an optional file can contain a dictionary keyed by speaker ID. This assumes speaker IDs are used consistently through all documents in that section. Appendix B contains a complete description of the file format.

Sections are managed individually as part of a larger repository of the whole Danish Gigaword Corpus. A validation script helps make sure that the sections comply with the file format.

4.2 Data protection

The corpus does not contain "sensitive" data as per the GDPR definition; that means no information identifying sexual orientation, political beliefs, religion, or health connected with utterer ID. This is achieved by stripping utterer information from social media content. Thus, data discussing potentially personally sensitive topics, for example, social media around political discussions, is disconnected from personally-identifying information. Further, social media content is supplied not as plain text but as IDs and code for rehydration, a process where the content is re-downloaded, thus avoiding redistribution of this content and affording

	Date	Form	Domain	Dialect	Socioeconomic status	Size (M)
Legal						**308.8**
Retsinformation	contemporary	written	Laws	legal	high	188.4
Skat.dk	contemporary	written	Tax code	legal	high	52.8
H-Sø	contemporary	written	Court cases	mixed	mixed	67.6
Social Media						**261.4**
Hestenettet	contemporary	written	forum	mixed	mixed	228.9
General Discussions	2 019 - 2 020	written	Twitter	mixed	mixed	32.0
Parliament Elections	2 019	written	Twitter	mixed	mixed	0.5
Conversation						**239.4**
OpenSubtitles	contemporary	spoken	Movie subtitles	mixed	mixed	130.1
Folketinget	2 009 - 2 019	spoken	Debates	rigsdansk	high	60.6
Europarl	2 004 - 2 008	spoken	Debates	standard	mixed	47.8
Spontaneous speech	2 019	spoken	Conversation	mixed	mixed	0.7
NAAT	1930 - now	spoken	Speeches	rigsdansk	high	0.2
Web						**101.0**
Common Crawl	contemporary	written	Web	mixed	mixed	101.0
Wiki & Books						**92.2**
Wikipedia	2 019 - 2 020	written	Encyclopaedic	standard	mixed	55.6
Danish Literature	1 700 - now	written	Literature	standard	mixed	25.6
Gutenberg	1 700 - now	written	Literature	standard	mixed	3.2
WikiBooks	2 019 - 2 020	written	Manuals	standard	mixed	2.6
WikiSource	1 700 - now	written	Literature	standard	mixed	2.5
Johannes V. Jensen	-	written	JVJ's works	rigsdansk	unknown	2.1
Religious texts	-	written	Religious	rigsdansk	unknown	0.6
News						**40.0**
TV2R	2 015 - 2 019	written	News	rigsdansk	high	10.0
DanAvis	1 999 - 2 003	written	News	rigsdansk	medium	30.0
Other						**1.2**
Dasem data[3]	contemporary	written	Other	mixed	mixed	0.7
Botxt	contemporary	written	Other	Bornholmsk	mixed	0.4
DDT	contemporary	written	Other	mixed	mixed	0.1
Sønderjysk	contemporary	written	Sønderjysk	Sønderjysk	mixed	0.02
TOTAL						**1 045**

Table 1: Text dimensions by text source in the Danish Gigaword corpus. Size in millions of words.

social media users the ability to delete their content without it being preserved by Danish Gigaword.

4.3 Test/Train partitions

Following the result that fixed test/train splits lead to unreliable results (Gorman and Bedrick, 2019), we avoid setting explicit test/train partitions in Danish Gigaword. We encourage users to select multiple random test splits. Since the Danish Gigaword is highly diverse, selecting multiple random splits will result in test sets with different biases following best practices (Søgaard et al., 2021).

4.4 Licensing

All corpus parts are licensed openly, for free distribution. We implement this with a mixture of Creative Commons general license (CC0) and CC-BY. Some older corpora (e.g., Kromann et al. (2003)) used the right under Danish copyright law to cite small excerpts of up to 250 words from published articles. While this is a creative solution to sharing digital language data, Danish Gigaword uses almost exclusively whole articles, as they are easier to work with, providing full context.

5 Distribution and sustainability

As mentioned earlier in this paper and by Kirkedal et al. (2019); Kirchmeier et al. (2019, 2020), one problem that plagues Danish NLP is a lack of large accessible corpora. To address this and maintain strict licensing standards that permit open and free redistribution, Danish Gigaword Corpus is hosted and freely distributed via `https://gigaword.dk/`. Alternative downloads will be provided through

major dataset distribution services at each significant release.

DAGW is an intrinsically open project. In a bid to improve and uphold its relevance at a broad level, the current group of participants covers academia, industry, and the public sector. However, the DAGW project is also volunteer-led and volunteer-driven, which brings intrinsic risk. Aside from cross-sector involvement, the DAGW project attempts to mitigate that risk through licensing, distribution, membership, community, and data integrity policies.

Strategically, the corpus strives for an improved balance. The contents in the first release, with this paper, reflect the data that is available in Denmark. Data that is legally required to be open and unlicensed dominates the corpus, reflecting the current state of text sharing in Denmark. We hope that this will become less conservative over time and particularly look forward to further donations of newswire and literature, so that NLP for Danish can start to offer Danish speakers improved technology.

The data is licensed CC-BY and CC0, which gives it broad reach and applicability, and makes it easier for stakeholders to join than copyleft or non-commercial licenses, such as GPL or CC-NC, would. It also improves distribution prospects: because of this licensing choice, DAGW can be hosted at a third-party research data repository like Zenodo or Figshare, shifting the responsibility for data hosting and provision to specialized third parties. The DAGW project also maintains an open policy, with any qualified stakeholders welcome to join, especially if there is a compatible donation of data. Denmark's size helps keep a manageable community. The Danish Gigaword also fosters community involvement by publishing results – for example, this paper. Finally, a small toolkit is included in the project's Github repository for automatic validation of any committed data, ensuring content integrity, quality, and uniformity.

6 Conclusion and Future Work

In Denmark, natural language processing is nascent and growing faster and faster. Content restrictions and conservative licensing abound. This paper presents the Danish Gigaword Corpus, a unified effort across many institutions and many Danish speakers to construct a billion-word corpus representing the language. It aims to be useful to a maximally broad and diverse group of users.

The Danish Gigaword Corpus is an active project. There is continuing effort to add sources that enhance the corpus' breadth, including fiction, older works from the 1800s, and newswire. DAGW continues past the first billion words, with data always released under Creative Commons license and freely distributed via `https://gigaword.dk/`.

We hope that this concrete and significant contribution benefits anyone working with Danish NLP or performing other linguistic activities and encourages others to publish language resources openly.

Acknowledgments

This work was not supported by any funded project or university initiative, but rather was a labour of love by the first two, *"fremmedarbejder"*, *"tosprogede"* authors, who thought Denmark really ought to have a decent-sized open corpus of Danish. And now it has. We are extremely grateful for the generous contributions of time, effort, and data from so many that made this project possible.

References

Douglas Biber. 1993. Representativeness in Corpus Design. *Literary and Linguistic Computing*, 8(4):243–257.

Christos Christodouloupoulos and Mark Steedman. 2015. A massively parallel corpus: the bible in 100 languages. *Language Resources and Evaluation*, 49(2):375–395.

Leon Derczynski, Kalina Bontcheva, and Ian Roberts. 2016. Broad Twitter Corpus: A diverse named entity recognition resource. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1169–1179, Osaka, Japan. The COLING 2016 Organizing Committee.

Leon Derczynski and Alex Speed Kjeldsen. 2019. Bornholmsk natural language processing: Resources and tools. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 338–344, Turku, Finland. Linköping University Electronic Press.

Christina Dideriksen, Riccardo Fusaroli, Kristian Tylén, Mark Dingemanse, and Morten H Christiansen. 2019. Contextualizing conversational strategies: Backchannel, repair and linguistic alignment in spontaneous and task-oriented conversations. In *Proceedings of the 41st Annual Conference of the Cognitive Science Society*, pages 261–267. Cognitive Science Society.

Jonathan Dunn and Ben Adams. 2020. Geographically-Balanced Gigaword Corpora for 50 Language Varieties. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2521–2529, Marseille, France. European Language Resources Association.

Xavier Ferrer, Tom van Nuenen, Jose M. Such, and Natalia Criado. 2021. Discovering and Categorising Language Biases in Reddit. In *Proceedings of the 15th International Conference on Web and Social Media*.

Riccardo Fusaroli, Bahador Bahrami, Karsten Olsen, Andreas Roepstorff, Geraint Rees, Chris Frith, and Kristian Tylén. 2012. Coming to terms: Quantifying the benefits of linguistic coordination. *Psychological Science*, 23(8):931–939. PMID: 22810169.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.

Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. Building large monolingual dictionaries at the Leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 759–765, Istanbul, Turkey. European Language Resources Association (ELRA).

Kyle Gorman and Steven Bedrick. 2019. We need to talk about standard splits. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2786–2791, Florence, Italy. Association for Computational Linguistics.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English Gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34.

René Haas and Leon Derczynski. 2021. Discriminating Between Similar Nordic Languages. In *Proceedings of the Workshop on NLP for Similar Languages, Varieties and Dialects*.

Dirk Hovy, Anders Johannsen, and Anders Søgaard. 2015. User review sites as a resource for large-scale sociolinguistic studies. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 452–461, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. 2018. Loss in Translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2984, Brussels, Belgium. Association for Computational Linguistics.

Sabine Kirchmeier, Peter Juel Henrichsen, Philip Diderichsen, and Nanna Bøgebjerg Hansen. 2019. *Dansk sprogteknologi i verdensklasse*. The Danish Language Council.

Sabine Kirchmeier, Bolette Pedersen, Sanni Nimb, Philip Diderichsen, and Peter Juel Henrichsen. 2020. World class language technology - developing a language technology strategy for Danish. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3297–3301, Marseille, France. European Language Resources Association.

Andreas Kirkedal, Barbara Plank, Leon Derczynski, and Natalie Schluter. 2019. The Lacunae of Danish Natural Language Processing. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 356–362, Turku, Finland. Linköping University Electronic Press.

Alex Speed Kjeldsen. 2019. Bornholmsk Ordbog, version 2.0. *Mål og Mæle*, 40. årgang:22–31.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit*, volume 5, pages 79–86.

Matthias T Kromann, Line Mikkelsen, and Stine Kern Lynge. 2003. Danish Dependency Treebank. In *Proc. TLT*, pages 217–220.

Anders Edelbo Lillie, Emil Refsgaard Middelboe, and Leon Derczynski. 2019. Joint rumour stance and veracity prediction. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 208–221, Turku, Finland. Linköping University Electronic Press.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Marianne Mortensen. 2016. Den bornholmske dialekt dør–og hvad så? Technical report, Roskilde Universitet.

Bolette Sandford Pedersen, Jürgen Wedekind, Sabine Kirchmeier-Andersen, Sanni Nimb, Jens-Erik Rasmussen, Louise Bie Larsen, Steen Bøhm-Andersen, Peter Henriksen, Jens Otto Kjærum, Peter Revsbech, Hanne Erdman Thomsen, Sanne Hoffensetz-Andresen, and Bente Maegaard. 2012. *Det danske sprog i den digitale tidsalder*. Springer.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring

the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Anders Søgaard, Sebastian Ebert, Jasmijn Bastings, and Katja Filippova. 2021. We Need to Talk About Random Splits. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, Kiev, Ukraine. Association for Computational Linguistics.

Steinþór Steingrímsson, Sigrún Helgadóttir, Eiríkur Rögnvaldsson, Starkaður Barkarson, and Jón Guðnason. 2018. Risamálheild: A very large Icelandic text corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Kristian Tylén, Riccardo Fusaroli, Pernille Smith, and Jakob Arnoldi. 2016. The social route to abstraction. *Cognitive Science*.

A Detailed corpus description

Here we detail some of the sections included in the corpus, specifying what they bring to the dataset to make it a rich resource covering a wide range of lexical, syntactic, and sociolinguistic phenomena expressed by Danish users. Table 1 provides an overview of the corpus.

A.1 TV2 Regionerne

This section is a contemporary Danish newswire sample: approximately 50 000 full newswire articles published between 2010 and 2019. It contains articles of regional interest, written following editorial standards. This section's value is in both its temporal variation, covering a decade of events, and its spatial variation, covering many local events across most of Denmark (TV2 Bornholm is excluded). As a result of local event coverage, the section contains many locally relevant named entities, which might otherwise not be present in a dataset of national news.

A.2 Folketinget

The Danish parliament (Folketinget) keeps a record of all meetings in the parliament hall.[4] All records have a transcript produced by commercial Automatic Speech Recognition (ASR) followed by post-editing by linguists employed by Folketinget for intelligibility, i.e., edit out dysfluencies, restarts, repairs, and mistakes. The transcript is, therefore, not a representation of spoken Danish but rather information content.

[4]There are no records of committee meetings or *samråd*.

In the parliament hall, one speaker at a time addresses members of the parliament. Monologues may include rebuttals or other comments to statements in previous monologues. While speakers can read aloud from a prepared statement or speak extemporaneously, we expect no difference to be apparent in the data because of the post-editing.

The Folketinget section covers parliament hall sessions between 2009 and 2019. It contains discussions on a wide range of topics, issues, and named entities relevant to Danish society.

A.3 Retsinformation

The site retsinformation.dk provides access to Danish laws and regulations and documents from the Danish parliament (Folketinget). The text is provided by Folketinget, ministries, the ombudsman of Folketinget, and Rigsrevisionen. The legislative texts in this section include a variety of features: Uppercase text, redaction where names and addresses are left out, itemized text with chapter and section numbering, headlines, words with intra-letter spacing.

A.4 Spontaneous speech

The conversational corpus included originates from interdisciplinary research conducted within the Interacting Minds Center,[5] and the Puzzle of Danish project[6] at Aarhus University. Transcribed Danish speech is generally a rare kind of data, and spontaneous speech especially so; these manually transcribed conversations thus form a valuable resource. Spontaneous and pseudo-spontaneous conversations come from various contexts, e.g., getting to know each other, solving a puzzle together, or making joint decisions. The participants have agreed on releasing anonymized transcripts of their conversations. All conversations involve two speakers, sometimes conversing face-to-face, sometimes via a chat tool. Speech is transcribed post-hoc by native speakers. Studies published relying on this data include Fusaroli et al. (2012), Dideriksen et al. (2019), and Tylén et al. (2016).

A.5 Danish Wikipedia

This section comprises a dump of Danish Wikipedia[7], stripped of Wikipedia-specific markup. The content is collaboratively written by a broad

[5]http://interactingminds.au.dk

[6]https://projects.au.dk/the-puzzle-of-danish/

[7]https://dumps.wikimedia.org/dawiki/

range of authors and covers many specific articles that often do not exist in other languages. Most content has been roughly checked for syntactic and orthographic canonicity by editors of the Danish Wikipedia and is a rich source of region-specific named entities, often situated in full, fluent sentences. The content is reproduced verbatim in accordance with the GNU Free Documentation License.

A.6 Europarl

The Europarl Parallel Corpus (Koehn, 2005) contains proceedings of the European Parliament in 21 European languages that were automatically extracted and aligned. We include the Danish part of the Europarl corpus and perform no pre-processing other than file format conversions.

A.7 OpenSubtitles

OpenSubtitles[8] is a website where a community writes and shares subtitles for mostly big-budget movies. We extract the Danish subtitles from the OpenSubtitles section of OPUS (Lison and Tiedemann, 2016). We clean the corpus to fix issues such as the capital letter I instead of the lower case letter L. We remove files that do not contain any characters specific to Danish (i.e., any of the letters *å*, *æ*, or *ø*).

A.8 Religious text

This section contains a Danish translation of the Bible from the Massively Parallel Bible corpus (Christodouloupoulos and Steedman, 2015) without any pre-processing other than file format conversion. We continue to look for other sources of religious textual content to improve the coverage and significance of this section.

A.9 Danish Twitter

Social media content is rich in unedited text, allowing for a very broad range of expressions. We know that social media users typically vary their language use to afford some representation for what would typically be communicated non-verbally, and while there are corpora for this for e.g. English, there are very few published corpora containing Danish social media text (e.g., (Hovy et al., 2015; Lillie et al., 2019)). This section contains two datasets of Danish tweets as dehydrated content, and includes a script for rebuilding this part of the corpus, thus permitting GDPR-compliant redistribution. The first dataset contains approximately 29 000 tweets in Danish from the #dkpol hashtag collected during the national parliamentary elections of 2019. The second dataset, consisting of approximately 1.6 million Danish tweets collected between April-June 2020, is not constrained by topic as tweets were collected using the 250 highest frequency Danish words.

A.10 DanAvis20

Corpus DanAvis20 consists of articles from various national Danish (daily) newspapers, including Aktuelt, Berlingske Tidende, Dagen, and Weekendavisen. The articles were published during 1999-2003. All texts included have been cleared for distribution under the CC0 license (cf. Section 4.4). As part of the clearing agreement, the papers were slightly edited by limiting all text quotes to 200 words (at most), picking sentences from longer papers at random. Sentences were mildly scrambled (DanAvis20 has no instances left of 4 adjacent sentences). Proper names were pseudonymized (except "Denmark", "København", "USA", and a few others). Infrequent content words (10ppm or less) were replaced in situ by "statistical cognates", i.e., words of similar frequency and equivalent morphosyntactic form (e.g., replacing "Der er sardiner i køleskabet." with "Der er skilsmissesager i forsikringsselskabet." while keeping "Ministeren rejser hjem igen"). As overall statistical and lexical properties of DanAvis20 are thus kept invariant, the corpus still provides good material for most NLP training purposes.

A.11 The *Bornholmsk Ordbog* Dictionary Project

Fictional texts of various kinds written in Bornholmsk, the dialect spoken on the Danish island of Bornholm,[9] have been digitized (OCR'ed and proofread) by volunteers working within the recently resumed *Bornholmsk Ordbog* dictionary project (Kjeldsen, 2019). Most of the material included is written by Otto J. Lund in the period 1930-48 (novels, short stories, and poems). The Bornholmsk subcorpus, which in its present state amounts to circa 400 K words, also includes folk stories published by J. P. Kuhre in 1938, and by K. M. Kofoed in 1935, fictional letters by various

[8] https://www.opensubtitles.org

[9] The language code for Bornholmsk under IETF BCP-47 is da-bornholm.

authors published in the 1930s, as well as poems by Alfred Jensen published in 1948 and various other texts from the same period. The non-standardized orthography varies considerably from source to source. The Bornholmsk part of the Danish Gigaword is a significantly extended dataset, well beyond that studied in earlier NLP work on the dialect (Derczynski and Kjeldsen, 2019).

B File format

The philosophy is to present data as plaintext, UTF8, one file per document. Accompanying metadata gives information about (for example) the author, the time or location of the document's creation, an API hook for re-retrieval of the document, among others.

B.1 Corpus Sections

As the corpus many sections, per section, we do the following:

- Give each corpus section a directory with an agreed name.
- Keep all plaintext as one file per document.
- Use a section prefix, underscore, and document identifier as the filename, e.g., "tv2r_01672".
- Do not use file extensions for the text files.
- Maintain a one-record-per-line JSONL file in the directory, with the same name as the section, and with "jsonl" suffix, e.g., "tv2r.jsonl". The content of this file should follow the JSONL format, see http://jsonlines.org.
- Each document's metadata is placed as a single JSON record in the JSONL metadata file, with a key "doc_id" matching the filename it describes. Separate entries by line breaks (i.e., one JSON object per line).
- A `LICENSE` file should be included in each section, stating the license under which the section is distributed. CC and public domain only! Preferably CC0 or CC-BY; CC-NC if we have to. No copyleft licenses - they restrict the use of the data too much, which we are trying to avoid.

Here are the fields for the standoff JSONL metadata file entries:

- `doc_id`: a string containing the document ID, which is also its filename. Begin with the section prefix, followed by an underscore. `String`. **Required**.
- `date_published`: the publication date of the source document, including the timezone. If only the year is available, use `year_published` instead. In the Python `strftime()` format, use `"%c %z"`. `String`. *Preferred.*
- `uri`: the URI from which the document originated; can be an API endpoint that links directly to the data. `String, URI`. *Preferred.*
- `year_published`: the year CE that the source document was published. `Integer`. Use only as an alternative to `date_published`. *Optional*.
- `date_collected`: the date at which the source document / API result collection, including the timezone. In the Python strftime() format, use `"%c %z"`. `String`. *Optional*.
- `date_built`: the date this document was included in the current version of the dataset, including the timezone. In the Python strftime() format, use `"%c %z"`. `String`. *Optional*.
- `location_name`: the name of the location of the document's origin. `String`. *Optional*.
- `location_latlong`: latitude and longitude of the document's origin. `List of two floats`. *Optional*.

B.2 Speech transcripts

To represent speakers in the text files, prefix each turn with "TALER 1:" (substituting whatever ID is appropriate). Note: there is no space before the colon; use one space after the colon. It is also OK to include the speaker's name directly if this is publicly known, e.g., "Thomas Helmig:".

For multi-speaker corpus sections, an optional `talere.jsonl` file can be included in the section, containing one JSON dictionary keyed by speaker ID. Speaker IDs should be consistent through all documents in a section. Speaker IDs need only be unique to speakers in a section, not universally.

Paper II

DaNewsroom: A Large-scale Danish Summarisation Dataset

Daniel Varab and **Natalie Schluter**
IT University of Copenhagen
Copenhagen, Denmark
{djam, natschluter}@itu.dk

Abstract

Dataset development for automatic summarisation systems is notoriously English-oriented. In this paper we present the first large-scale non-English language dataset specifically curated for automatic summarisation. The document-summary pairs are news articles and manually written summaries in the Danish language. There has previously been no work done to establish a Danish summarisation dataset, nor any published work on the automatic summarisation of Danish. We provide therefore the first automatic summarisation dataset for the Danish language (large-scale or otherwise). To support the comparison of future automatic summarisation systems for Danish, we include system performance on this dataset of strong well-established unsupervised baseline systems, together with an oracle extractive summariser, which is the first account of automatic summarisation system performance for Danish. Finally, we make all code for automatically acquiring the data freely available and make explicit how this technology can easily be adapted in order to acquire automatic summarisation datasets for further languages.

Keywords: automatic text summarisation, data collection, danish corpus

1 Introduction

Dataset development for automatic summarisation systems is notoriously English-oriented. This is surprising. On the system user-side, a more feasible access (for example, summaries) to the increasing amounts of digital information informing daily life is of inherent interest to potential users across the globe. At the same time, automatic summarisation provides a challenging NLP test-bed to investigate the limits of deep learning for NLP, and for downstream evaluation of basic core NLP tasks like discourse analysis, co-reference resolution, and other types of parsing (Lee et al., 2018; Li et al., 2016). Yet, only very limited datasets exist in languages other than English (Nguyen and Daumé III, 2019; Schluter and Martínez Alónso, 2016).

By *automatic summarisation dataset* we denote a collection of entire documents each paired up with at least one manually written summary; the summaries of such a dataset are intended as a summaries for those documents and not as headlines, or a list of facts or highlights. In fact, until recently central larger-scale automatic summarisation datasets have not included been composed of any summaries. Namely, Rush et al. (Rush et al., 2015) were the first to recast the English Gigaword dataset (Parker et al., 2011; Napoles et al., 2012) as a headline-type large-scale summarisation generation dataset. And the CNN/Daily Mail dataset (Moritz Hermann et al., 2015), a question answering dataset, was first recast by (Cheng and Lapata, 2016; Nallapati et al., 2016) as an automatic summarisation dataset. These two datasets have been central to more recent automatic summarisation system development.

Headline and highlights datasets are are not ideal for the development of summarisation systems, but because of their scale and in the absence of alternatives, they provided a much needed crucial prerequisite for neural system development.

The advent of the English language Newsroom dataset (Grusky et al., 2018)–a dataset of 1.3 million English article-summary pairs that was created by collecting manually written summaries from news articles provided the first large scale first-class summarisation dataset. To our knowledge, it is also the only existing large-scale automatic summarisation dataset, prior to this paper. With this work, we adopt, extend, and extensively describe an approach to automatically constructing a Danish language automatic summarisation dataset. This essentially (1) provides the first Danish language automatic summarisation dataset, (2) enables neural system development for Danish under a monolingual setting, and establishes (3) the first non-English large-scale automatic summarisation dataset.

Our contributions. With this paper, we contribute the following.

- We establish the **first automatic summarisation dataset for Danish**.
- By contrast to other non-English languages, where new dataset development have been rather limited (i.e., less than 2K document-summary instance pairs) if existent, our dataset, DaNewsroom, is *large-scale*, with more than 1.1 million document-summary instance pairs surviving our quality-control filters. This means, we are presenting **the first non-English large-scale dataset** curated and quality-controlled specifically automatic summarisation system development.
- We **adopt and make key extensions to Grusky et al.'s (2018) methodology** for the development of their Newsroom dataset to the Danish language. In particular, our clarifications, extensions, and associated code presented here **permit researchers to easily develop similar automatic summarisation datasets for other non-English languages**.
- We present the **first account of baseline performance for Danish automatic summarisation** as a point of reference for future neural systems.

We make the code for generation of the dataset, the baseline systems, as well as the dataset itself publicly available[1].

2 Current central datasets for automatic summarisation

We now survey the central datasets for automatic summarisation system development and benchmarking. By "central dataset for automatic summarisation", we mean that the dataset (1) is not a specialised type of summarisation exclusive to a particular domain (like scientific article abstract generation), and that it (2) is typically used in automatic summarisation system benchmarking. All central datasets today are composed entirely of English news articles-summary pairs.

DUC 2004. The DUC2004[2] is currently the most central dataset for automatic summarisation system benchmarking. This is a manually curated multi-document summarisation dataset, whose instance pairs consist of sets of hand-picked, highly related documents paired with summaries about that set of documents, written specifically for the construction of this dataset by different writers.

Despite the added task dimension of having sets of multiple documents to summarise, rather than one single document to summarise, all current state-of-the-art systems to the authors' knowledge first concatenate these multiple documents together into a single document and then summarise the whole as though it were a single document.

Multiple summaries, on the other hand, were meant to provide a more accurate, less author-biased, gauge of system output quality (Nenkova and Passonneau, 2004), by averaging relevant metrics over reference summaries written by different authors.

The DUC 2004 is very small and unsuitable for supervised machine learning in general; as such, it has been primarily used for unsupervised automatic summarisation, and more recently as a test set for neural automatic summarisation systems. The DUC 2004 dataset contains 30 document set-summary set pairs, with an average summary length of 665 bytes/100 words.

CNN/Daily Mail. The CNN/Daily Mail dataset is an automatically generated dataset constructed by crawling cnn.com and dailymail.co.uk. It was originally introduced as a Question Answering dataset (Moritz Hermann et al., 2015) and comprises articles accompanied by information boxes of a couple of bulleted article highlights. These articles were later converted into a summarisation dataset[3] (Cheng and Lapata, 2016; Nallapati et al., 2016) by considering the bullet points as a description of the article and concatenating the listed facts into a single summary. The summaries have on several accounts been described as being highly extractive (Grusky et al., 2018; Chen et al., 2016). The dataset contains over 312K articles of mean length 781 words, accompanied by summaries with mean length 56 words.

Though a useful adaptation, this method for the automatic creation of a new summarisation dataset has two major flaws. First, as we discussed in Section 1, bullet-point highlights are not manually written summaries of articles, and are therefore system development over this dataset does not exactly automatic summarisation system development. Second, data collection is restricted to news outlets who collect highlights in information boxes within news articles. As such, the data collection strategy doesn't correspond to conventional document structure that is generalised across a wide range of news outlets.

Newsroom. Newsroom (Grusky et al., 2018) is a large-scale dataset created by conducting a scrape of news articles from 38 English language news outlets covering the period 1997-2018. The scrape was enabled by the The Internet Archive (archive.org), a non-profit organisation which provides a platform for hosting and accessing past published internet content. Together with archive.org, this work takes advantage of the use of the Semantic Web[4] and properties of Facebook's Open Graph protocol[5] which encouraged online publishers to insert a special metadata summary for each news article. The dataset contains 1.3 million document-summary pairs, with articles of mean length 659 words and mean summary length 27 words.

3 Towards a DaNewsroom

We extend the work of Newsroom (Grusky et al., 2018) and use The Internet Archive[6], a non-profit archiver. Specifically, we use the Wayback Machine[7], a sort of automatic archive system and a product of The Internet Archive. The Wayback Machine has automatically and systematically scraped the internet for the past 20 years. As such, the Wayback Machine provides the history of the web through snapshots and has since collected more than 300 billion websites–all directly accessible through their own online databases. The Wayback Machine provides an API[8] that enables users to query their databases with URLs[9]. Since this historical content is also freely available through the endpoint web.archive.org/web/TIMESTAMP/URL we are equipped with a method to retrieve web content across time, and in this case, news articles from the past. It is with this procedure that Newsroom was collected. Though the process of the acquiring URLs was not described by (Grusky et al., 2018), we provide a reproducible approach with accompanying code for retrieving URLs[10] here.

3.1 Danish News Sites

To construct a dataset from web crawls we first curate a list of sites that will act as search strings for our queries to the Wayback Machine API. This provides the URLs to the stored snapshots hosted in the The Internet Archive's databases. Unlike the English Newsroom, where Grusky

[1] github.com/danielvarab/da-newsroom
[2] https://duc.nist.gov/duc2004/
[3] github.com/abisee/cnn-dailymail
[4] w3.org/standards/semanticweb
[5] ogp.me
[6] archive.org
[7] archive.org/web
[8] web.archive.org/cdx/search/cdx
[9] see documentation at github.com/internetarchive/wayback/tree/master/wayback-cdx-server
[10] github.com/danielvarab/da-newsroom

et al. (2018) used an already curated list of appropriate English language news URLs, no such extensive curated list of Danish media exists, and the Danish Wikipedia[11] only lists nine outlets.

We extend the list from Wikipedia and compose a list of news outlets that are (1) well-known, (2) have existed for the past 20 years, and (3) are included by the Wayback Machine. While the Wayback Machine hosts snaphots of the entire web over time, and in theory across all languages, through manual inspection of coverage of non-English sites it becomes apparent that snapshots are biased towards English sites. A central challenge is therefore the sparse coverage of Danish websites. We list the sites we collect URLs from in Table 1.

DOMAIN	NEWS OUTLET TYPE
altinget.dk	political news outlet
avisen.dk	local news outlet
berlingske.dk	national news outlet
borsen.dk	financial news outlet
bt.dk	tabloid
dagens.dk	local news outlet
dr.dk	national news service
ekstrabladet.dk	tabloid
finans.dk	financial news outlet
fyens.dk	local news outlet
gaffa.dk	music news outlet and blog
ing.dk	tech and science outlet
jyllands-posten.dk	news outlet
kristeligt-dagblad.dk	national news outlet
lokalavisen.dk	collection of local news outlets
nyheder.tv2.dk	national news service
seoghoer.dk	tabloid
version2.dk	tech outlet and blog
videnskab.dk	pop science outlet

Table 1: Danish news sites from which URLs are collected.

We carry out extensive filtering of article-summary pairs based on URL and document contents heuristics (Cf. Sections 3.2 and 5). Figures 1 and 2 show the resulting distribution of article-summary pairs based on domain name and year of publication, respectively.

3.2 Obtaining URLs

Using the list of news sites found in Table 1, we query the Wayback Machine API for URLs. Scraping a domain d is in its most basic form done by calling the archive.org endpoint[12] with the HTTP parameters `url=`d and `matchType=domain`. The `url` parameter acts as a query and specifies a target site, while `matchType` defines which snapshots the query matches (i.e., exact query matches vs. site-match). In addition to these two parameters we use two additional HTTP parameters; `collapse`[13] and `filter`[14]. This removes duplicated URLs and filters out resource/error snapshots. Note that collapsing and filtering also can be done post hoc. We refer to the API documentation for further details of each parameter[15].

This strategy for obtaining URLs produces 14 million URLs snapshots going back 20+ years. A great deal of URLs are, however, of poor quality. In addition, at this large a scale, due to slow download rates from the free archive.org/web service, scraping all possible urls is unfeasible. Therefore detecting noisy (poor quality) urls can help reduce the risk of wasting download time on unusable articles. We therefore filter URLs according to two simple heuristic guidelines.

1. Extract, if any, extension for each URL and prune all instances that contain extensions of common assets such as javascript, stylesheets, fonts, and image files (js/css/tff/png etc.). Most cases of this should be caught by the above *mimetype* filter, however, it only applies to websites that follow conventions and use the appropriate mimetype.

2. Prune URLs that contain the regular expression `(-[a-zA-Z]+){3,}`. This effectively matches URLs that contain three alphabetic sequences delimited by three dashes. We motivate this by the best-practise naming, human-readable URLs (aka "hURLs"), which is a common URL-schema for news outlets that suggests article URLs align with the corresponding article title: for example, *berlingske.dk/samfund/derfor-er-det-saa-svaert-at-vaelge-kampfly*. An example of a URL that is filtered out is *dagens.dk/arkiv/Politik?page=476*. We inspect the results manually and observe a noticeable reduction of unusable pages such as front pages, and web assets.

These two filters reduce the initial 14 million URLs to about 4.8 million, before any document-intrinsic quality control filters (Cf. Section 5).

3.3 Scraping Articles

With a hURL-filtered collection of about 4.86 million candidate URLs we scrape the content found at the end of each candidate URL hosted by the Wayback Machine. We use the Newsroom[16] Python package provided by Grusky et al. (2018) to download articles as well as extract the contents. The package enables concurrent downloads to a compressed format (jsonl+gzip). This is a straight forward, but time consuming process. Downloading documents from a single machine with the default configuration, downloads a mere 1-3.5 articles per second, with frequent stalls and fluctuating download speeds. The final scrape of DaNewsroom took more than a week to finish and resulted in about 3.59 million downloaded articles, a reduction of 26% compared to number URLs initially provided. These lost articles

[11] da.wikipedia.org/wiki/Aviser_i_Danmark#Landsd%C3%A6kkende_dagblade

[12] web.archive.org/cdx/search/cdx

[13] *[...]&collapse=url*

[14] *[...]&filter=statuscode:200&filter=mimetype:text/html*

[15] github.com/internetarchive/wayback/tree/master/wayback-cdx-server

[16] github.com/lil-lab/newsroom

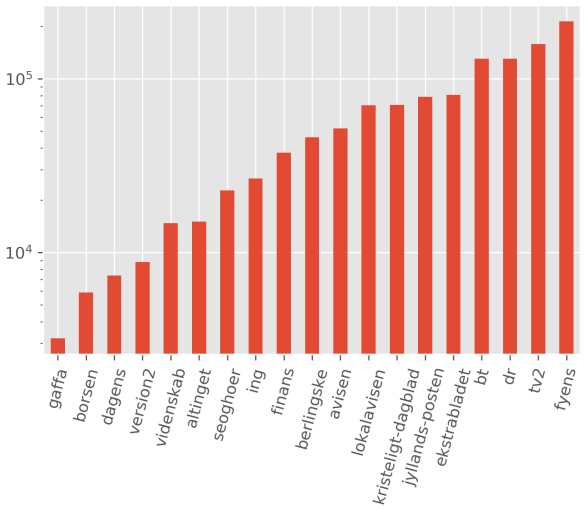


Figure 1: Article count (log-scale) in for each domain name in DaNewsroom.

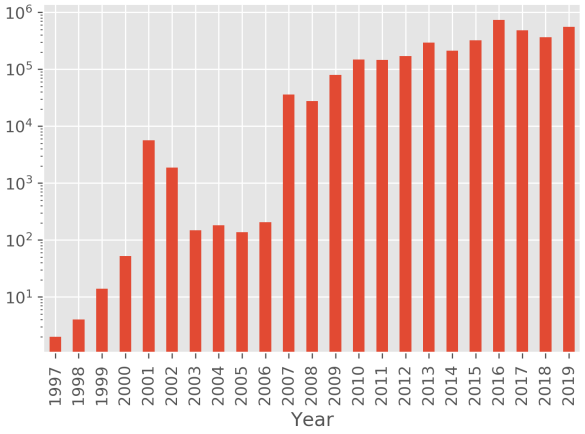


Figure 2: Distribution of articles across years for each collected site in DaNewsroom. The y-axes is plotted in log-scale to highlight the low presence of articles in the late 90's and early 2000's.

may be explained by server errors from the Wayback Machine which are caused by either snapshots not existing or especially lengthy request time-outs.

3.4 Extraction

For extracting samples from the downloaded articles we employ the newsroom-extract command-line tool from the Newsroom package. The package uses Readability[17] to retrieve the main article content and title, and uses SpaCy (Honnibal and Montani, 2017) for tokenisation to compute metrics of compression, coverage and density. The summaries are extracted if there is at least one out three metadata tags: *og:description*, *twitter:description* or *description*. When extracting, we discovered that Danish websites appear not to have embraced, or at least have been slow to adapt to the semantic web metadata tags for summaries. Tags are often present, but contain either empty strings, or site-wide descriptions that are not specific to the article at hand. As shown in Figure 2, there is a corresponding lack of of older articles in the dataset.

[17]pypi.org/project/readability-lxml

Since the Newsroom package is intended for English, we clone the repository and modify it to support multiple languages and in particular Danish tokenisation during extraction.

4 Document-Summary Descriptive Measures

To explore the quality and extractiveness of summaries with respect to documents, Grusky et al. (2018) carried out a series of measurements over *extracted fragments*: greedy n-gram overlaps between an article body and reference summary: coverage, density and compression. We present the definitions used by Grusky et al., and apply these same measures to DaNewsroom. We then propose using these measures as an automatic tool for identifying high-quality article-summary pairs.

Let (A, S) be a instance pair of an article $A = (a_1, a_2, .., a_n)$ and a summary $S = (s_1, s_2, .., s_m)$ consisting of tokens a_i and s_i respectively. And let $|A| := n$ and $|S| := m$.

Extractive Fragments. The set of *extractive fragments* $F(A, S)$ is the set of longest common sequences of tokens in A and S.

Coverage. Coverage measures the extractiveness of a summary–the extent to which the sequences of extractive fragments (the article) covers the the summary itself. As extractiveness increases, coverage tends towards 1. Conversely, as abstractive-ness increases and novel words are introduced, coverage tends towards 0.

$$\textsc{Coverage}(A, S) = \frac{1}{|S|} \sum_{f \in F(A,S)} |f| \tag{1}$$

The next measure takes this into consideration.

Density. Density is identical to coverage, except that the length of fragments (in the summary) is squared. This results in a measure that scores higher for summaries that contain long extractive fragments. If an abstractive summary contains random words from the article, it will also score high in coverage despite being abstractive. By contrast, because the extractive fragments are short, density will indicate extractive-ness.

Thus, combining density with coverage allows one to identify summaries that are mixed extractive and abstractive (so-called "mixed summaries") that compose abstractive-like summaries from short sequences of text found in the article.

$$\textsc{Density}(A, S) = \frac{1}{|S|} \sum_{f \in F(A,S)} |f|^2 \tag{2}$$

Compression. Compression expresses the compression rate of tokens between the article and the summary: the summary to document length ratio.

$$\textsc{Compression}(A, S) = |A|/|S| \tag{3}$$

5 Removing Low Quality Articles

After scraping and and extracting (Section 3), we are left with 3.6 million articles, of which the majority we expect to be of poor quality (given, in particular that Grusky et al. (2018) retained only 1.3 million of the original 100+ million articles). We introduce a few robust high-recall techniques to detect better quality instance pairs and improve the overall quality of DaNewsroom.

- First we removed articles having either empty summaries or article bodies. The portion of empty summaries are 12.1%, and 6.3% of the articles contained an empty body. We observe some overlap with union of the two being 15.5%.
- Secondly, we filter out articles where summaries and bodies are non-unique: if the entire summary or article body is present in more than one document we exclude it from the dataset. This constitutes 45.4% and 31.3% respectively, with a combined presence in almost half of all articles (49.9%).
- Third, we filtered document-summary pairs where the summary was of longer, equal or just slighter shorter length of the article body. Specifically, we filter out articles where $\textsc{compression}(A, S) < 1.5$. Future work could consider further tuning (and increase) of this threshold. We opted err cautiously (for high recall) and keep possibly less interesting samples in the dataset that future work can then filter out, rather than filter out perfectly valid samples (false negatives).

Table 2 summarises the reduction in document-summary pairs across the various stages of filtering. The result of these steps is DaNewsroom.

STAGE	COUNT	% REDUCTION
Filtered URLs	4,859,658	-
Downloaded Articles	3,590,150	73.88%
Post Extraction	3,578,679	73.64%
Basic Filtering	1,175,238	24.18%
Compression Cut-off	1,132,734	23.31%

Table 2: Article filters and the percentage of documents after the entire dataset.

6 Analysis of Measures over DaNewsroom

The above document-summary descriptive measures provide us with a feasible way to ascertain the "extractiveness" or "abstractiveness" of article-summary pairs. In Grusky et al. (2018) there is an emphasis on the signal expressed by the combination of *coverage* and *density* which is displayed in a bivariate plot. We generate a similar plot for DaNewsroom, in Figure 3. In addition we present the same density plot with an increased threshold of $\textsc{Density}(A, S) < 50$ in Figure 4. This new plot represents 98.6% of DaNewsroom in contrary to that of $\textsc{Density}(A, S) < 5$ (Figure 3) representing only 43.4% of DaNewsroom.

From Figure 4 we are able to see two clusters of articles. The top-right cluster is composed almost entirely of long extractive summaries: long extractive summaries will have high density. In the bottom left cluster, summaries contain longer spans, though not entire sentences, from the article body. Upon manual inspection of samples this cluster appears to be of particularly high quality.

In Figure 5 we see the compression distribution in DaNewsroom. Recall that compression represents to which extent the summary compresses the article body (token-wise). We observe that summary compression is distributed mainly below 20 followed by a steep long tail. This, together with the mean summary token count (20), tells us that we should not expect particularly long documents.

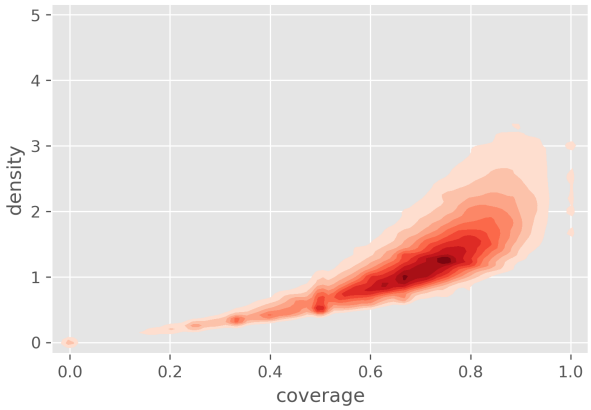


Figure 3: Density distribution where $\textsc{Density}(A, S) < 5$. The axes are the measures extractive fragment converage (x-axis) and density (y-axis) measures in DaNewsroom.

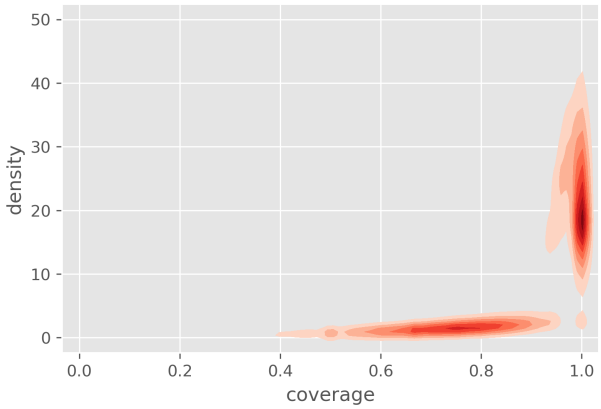


Figure 4: Plot displaying the dataset density between extractive fragment converage (x-axis) and density (y-axis) measures in DaNewsroom where $\textsc{Density}(A, S) < 50$.

In the appendix, we provide example article-summary excerpts to illustrated the different clusters of the distribution.

7 Getting DaNewsroom

We distribute DaNewsroom as a list of URLs which link to snapshots hosted at The Internet Archive. Together with the modified Newsroom command-line tool, one may reconstruct the dataset. We make the modified Newsroom package and build script freely available[18]. With this approach we hope to encourage extendability as the dataset

[18] github.com/danielvarab/da-newsroom

	DaNewsroom			Newsroom		
	R-1	R-2	R-L	R-1	R-2	R-L
LEAD-3	42.80	35.97	40.11	30.72	21.53	28.65
Oracle	90.13	81.40	90.13	88.46	76.07	88.46
TextRank	26.92	14.95	22.23	22.82	9.85	19.02
ICSISumm	26.83	14.99	22.22	-	-	-

Table 3: F_1-score ROUGE on the test set of Newsroom

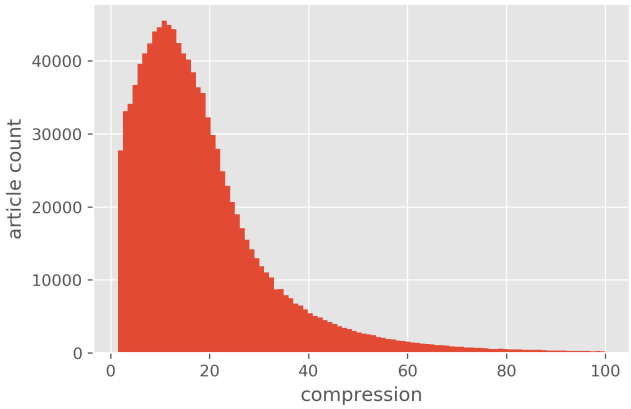


Figure 5: Compression distribution, clipped at 100, in DaNewsroom.

can be easily be extended as well as replicated to other languages following the same methodology.

We split the URLs across sites, grouping URLs by domain and split them into three sets (train/dev/test) over three steps: First we shuffle and split the group into a train, test and dev(elopment) set, with a 80/10/10 ratio. Then we merge all samples belonging to the same split (train, dev or test) and save them to separate files. See Table 4 for descriptive statistics of the dataset and splits.

split		count	$\lvert A\rvert$	a $\pm$ s ($\lvert S\rvert$)
train	(types)	2,733,973	-	-
	(tokens)	389,008,391	404.8	24.53±12.8
dev	(types)	738,883	-	-
	(tokens)	48,733,808	403.7	24.51±12.6
test	(types)	738,480	-	-
	(tokens)	48,674,409	407.1	24.52±12.7
full	(types)	3,499,762	-	-
	(tokens)	3,146,648	404.9	24.52±12.6

Table 4: Descriptive statistics of tokens and types in the splits of DaNewsroom. Average (a) for articles ($\lvert A\rvert$) and summaries ($\lvert S\rvert$) in addition to standard deviation (s) for summaries are over tokens counts only. The count column is over the set of all article-summary pairs ($\lvert A\rvert + \lvert S\rvert$) in the entire dataset.

8 Baselines

For comparison with future system performance, and since no prior work has been done previously on Danish summarisation, we now introduce report the performance for handful of simple but strong unsupervised baseline models (TextRank, ICSISumm, and Lead-3), together with an oracle extractive model (Fragment Oracle). See Table 5 for an overview of the model performances on DaNewsroom.

8.1 TextRank

TextRank (Mihalcea and Tarau, 2004) is an unsupervised extractive graph-based extractive summarisation system makes use of a version the PageRank algorithm (Page et al., 1999) to importance weight input document sentences for their selection into the output summary. Based on the words of the documents (for nodes) and the lexical similarity (for edges), a text network is formed and words obtain a centrality (of the network) weighting as a measure of their importance, and upon which sentence weighting depends. We use the implementation provided by the Python library Gensim[19] (Řehůřek and Sojka, 2010) which follows the recent extensions proposed by (Barrios et al., 2016). This is the exact system used by (Grusky et al., 2018), except that Gensim does not support custom tokenisation and sentence segmentation. Therefore, for this model, we employ English language tools.

8.2 ICSISumm

ICSISumm (Gillick and Favre, 2009) is an unsupervised summarisation system that generates extractive summaries, by outputting the set of input document sentences that globally and cummulatively contain the most important document concepts (bigrams). Bigram importance is approximated by bigram frequency in the input document set. We use the code associated with the paper[20]. We also include our code extensions in our own repository for reproducibility.[21]

8.3 Lead-3

Lead-3 copies the three first sentences of the article and presents directly them as the summary. The approach takes advantage of the fact that news articles often start with a paragraph that pitches the article. Despite the simplicity of this approach, Lead-3 is one of the strongest baselines for neural automatic summarisation (of online news articles), though it should serve, rather, as a type of lower bound and sanity check during system development.

8.4 Fragment Oracle

We include the Extractive Fragment Oracle as described in (Grusky et al., 2018). This model uses the fragments function $F(A, S)$ and composes a summary by concatenating

[19] radimrehurek.com/gensim/summarisation/summariser.html
[20] github.com/benob/icsisumm
[21] github.com/danielvarab/da-newsroom

	Extractive			Mixed			Abstractive		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
LEAD-3	60.95	59.38	60.69	26.38	13.07	20.31	16.80	3.85	11.57
Oracle	99.36	99.21	99.36	88.72	76.45	88.72	71.24	46.78	71.24
TextRank	34.07	23.00	29.47	22.10	8.15	16.66	15.11	2.73	10.73
ICSISumm	34.20	23.54	29.40	20.71	7.33	14.90	14.72	2.54	9.97

Table 5: F_1 ROUGE scores on three subsets of the development set. Extractive, mixed and abstractive are binned categories of the density measure. These cut-off values are taken directly from (Grusky et al., 2018).

the returned fragments of the function. This model, therefore, has access to the reference summary and acts as a upper bound for extractive methods. Surpassing the performance of this model would require an abstractive summarisation approach.

We note that the Fragment Oracle approach does not attempt to repair or rearrange fragments in any way, and merely concatenates the fragments in the order they are returned by $F(A, S)$. This often results in incoherent summaries that still score high ROUGE scores.

9 Baseline Evaluation

We use the standard ROUGE-1 (R-1), ROUGE-2 (R-2) and ROUGE-L (R-L) for evaluation, as it have been shown to be the ROUGE measures that are the most correlated measures with human judgements of summarisation (Hong et al., 2014). We leverage the Newsroom Python package which follows the default parameters for ROUGE. For word tokenisation for all systems, except TextRank, we use the Danish SpaCy tokeniser. TextRank uses the English tokeniser included in Gensim as it does not support custom tokenisation. We don't employ any lemmatisation. For sentence segmentation we use the English Gensim sentence segmenter.

For PageRank and ICSISumm, we must also input an output summary budget parameter. For PageRank, we employ a grid search, optimising for R-1, on the development set and find 35 to give us the best results. We adopt the same budget for the ICSISumm experiments.

In Table 3 we see the scores for all four models on the test set. We include scores reported in the Newsroom paper for relative comparison. Relatively, results on DaNewsroom follow the same trend as those reported on the Newsroom dataset. LEAD-3 and Oracle significantly outperform the other summarisation systems across both datasets. TextRank and ICSISumm are almost indistinguishable on all three ROUGE metrics on DaNewsroom, only differentiating by at most 0.1 absolute percentage point.

In Table 5 we see the scores produced by the four presented baseline models on three subsets of the development set. These subsets are binned categories of Density(A, S) values. We follow the cut-off values directly from (Grusky et al., 2018) of 1.5 and 8.1875, where $abstractive$ = Density(A, S) $<=$ 1.5, $mixed$ = 1.5 $>$ Density(A, S) $>$ 8.1875, and $extractive$ = Density(A, S) $>$ 8.1875. The distribution of these bins is given in Figure 6. Again, LEAD-3 and Oracle outperform the two remaining models by a large margin. On the extractive subset LEAD-3 jumps to an F_1-score of 60 across all ROUGE metrics, and our Oracle model pushes 100, due to the matching extractive character of the method. TextRank and ICSISumm both, as expected, improve in performance, most likely due to their being are purely extractive methods. Equally expected, these latter models score lower on the two other subsets: *mixed* and *abstractive*.

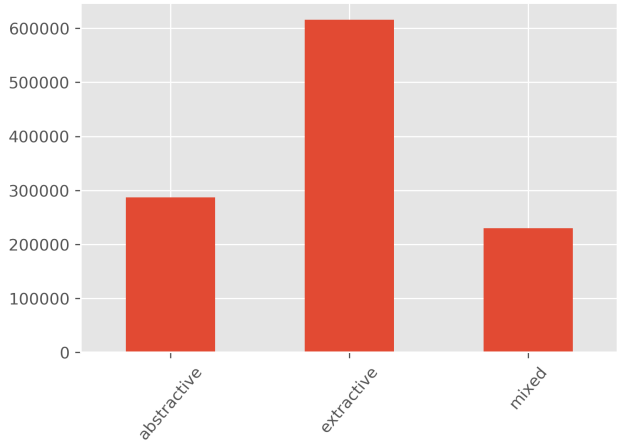


Figure 6: Distribution of binned categories (extractive, abstract or mixed) in DaNewsroom. About half of samples are categorised as extractive, while the remaining half is a mixture of abstractive and mixed samples.

10 Concluding remarks

We have presented the first Danish automatic summarisation dataset, which is also the first large-scale non-English for this task, together with baseline performance over the test sets. Dataset development for automatic summarisation systems has indeed been notoriously English-oriented. However, system performance problems related to automatic performance metrics required to gauge the performance of any realistic development of these systems for English itself, let alone other non-English languages is still problematic (Schluter, 2017), and could impede making the actual business case of automatic summarisation development. With this dataset, we are finally able to gain some understanding of the true performance of currently developed systems outside of the English arena. More over, we have provided explicit guidelines and tools to apply the same method to further languages.

11 Bibliographical References

Barrios, F., López, F., Argerich, L., and Wachenchauzer, R. (2016). Variations of the similarity function of textrank for automated summarization. *CoRR*, abs/1602.03606.

Chen, D., Bolton, J., and Manning, C. D. (2016). A thorough examination of the CNN/daily mail reading comprehension task. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2358–2367, Berlin, Germany, August. Association for Computational Linguistics.

Cheng, J. and Lapata, M. (2016). Neural summarization by extracting sentences and words. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 484–494, Berlin, Germany, August. Association for Computational Linguistics.

Gillick, D. and Favre, B. (2009). A scalable global model for summarization. In Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing, pages 10–18, Boulder, Colorado, June. Association for Computational Linguistics.

Grusky, M., Naaman, M., and Artzi, Y. (2018). Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. 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 708–719, New Orleans, Louisiana, June. Association for Computational Linguistics.

Hong, K., Conroy, J., Favre, B., Kulesza, A., Lin, H., and Nenkova, A. (2014). A repository of state of the art and competitive baseline summaries for generic news summarization. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), pages 1608–1616, Reykjavik, Iceland, May. European Language Resources Association (ELRA).

Honnibal, M. and Montani, I. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Lee, K., He, L., and Zettlemoyer, L. (2018). Higher-order coreference resolution with coarse-to-fine inference. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 687–692, New Orleans, Louisiana, June. Association for Computational Linguistics.

Li, Q., Li, T., and Chang, B. (2016). Discourse parsing with attention-based hierarchical neural networks. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 362–371, Austin, Texas, November. Association for Computational Linguistics.

Mihalcea, R. and Tarau, P. (2004). TextRank: Bringing order into text. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pages 404–411, Barcelona, Spain, July. Association for Computational Linguistics.

Moritz Hermann, K., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). Teaching machines to read and comprehend. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama and R. Garnett (Eds.), *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., pp. 1693–1701.

Nallapati, R., Zhou, B., dos Santos, C., GuÌ‡lçehre, Ç., and Xiang, B. (2016). Abstractive text summarization using sequence-to-sequence RNNs and beyond. In Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, pages 280–290, Berlin, Germany, August. Association for Computational Linguistics.

Napoles, C., Gormley, M., and Van Durme, B. (2012). Annotated gigaword. In Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction, AKBC-WEKEX '12, pages 95–100, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nenkova, A. and Passonneau, R. (2004). Evaluating content selection in summarization: The pyramid method. In Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004, pages 145–152, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.

Nguyen, K. and Daumé III, H. (2019). Global voices: Crossing borders in automatic news summarization. In Proceedings of the 2nd Workshop on New Frontiers in Summarization, pages 90–97, Hong Kong, China, November. Association for Computational Linguistics.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Parker, R., Graff, D., Kong, J., Chen, K., and Maeda, K. (2011). English gigaword fifth edition. Linguistic Data Consortium.

Řehůřek, R. and Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, pages 45–50, Valletta, Malta, May. ELRA. http://is.muni.cz/publication/884893/en.

Rush, A. M., Chopra, S., and Weston, J. (2015). A neural attention model for abstractive sentence summarization. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 379–389, Lisbon, Portugal, September. Association for Computational Linguistics.

Schluter, N. and Martínez Alónso, H. (2016). In Actes de la conférence conjointe JEP-TALN-RECITAL, volume 2, pages 349–354.

Schluter, N. (2017). The limits of automatic summarisation according to ROUGE. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 41–45, Valencia, Spain, April. Association for Computational Linguistics.

Appendix

In this appendix, we present five examples from DaNewsroom. Each article-summary pair displays different properties with respect to the measures described in Section 4. Each example belongs to one of binned categories (extractive, abstract or mixed) described in 9.

Figure 7 shows a *mixed summary*. We observe that the summary consists almost entirely of spans and tokens from the article body, but not an entire sentence. This is still an abstractive summary, but illustrates the special case where the summary to some degree is composed of spans from the article.

Summary: Windows 10 er på gaden, og har du Windows 7 eller en nyere version af styresystemet på din pc eller tablet, kan du opgradere gratis.
Start of Article: Nu er det længe ventede Windows 10 ankommet. 29. juli udkom det seneste i rækken af Microsofts styresystemer. Har du Windows 7, Windows 8 eller Windows 8.1 på din pc, kan du hente og installere det nye styresystem. Gratis. Du har et år (fra 29. juli) til at tage beslutningen, før der skal betales en opgradering, men på Datatid TechLife kan vi allerede nu fornemme, at Windows 10 bliver godt. [...]

Figure 7: Mixed summary which combines extractive spans to produce an abstractive descriptions of the article.

Figure 8 provides an example of an extractive summary. Here the exact summary is contained as the first sentence in the article. This is a well known tendency in news articles and provides evidence that the LEAD-3 baseline is well-motivated.

Summary: En international lufthavn i Florida har fredag aften været ramme for en skudepisode
Start of Article: En international lufthavn i Florida har fredag aften været ramme for en skudepisode Mindst fem mennesker har mistet livet i en skudepisode i den internationale lufthavn Fort Lauderdale-Hollywood, der ligger i staten Florida lige nord for byen Miami. Det oplyser det lokale politi. [...]

Figure 8: Extractive summary which is the first sentence in article body.

In Figure 9 we observe a challenging example of an abstractive summary which compresses and selectively filters information contained throughout the article. Details and names are removed, but the general theme of the article remains.

Figure 10 shows another abstractive summary that deviates entirely from the form of the article. This is also a prime example of an abstractive summary. The summary compresses the the salient information contained in the three first sentences into a single information rich sentence.

Summary: EU skal stå sammen om at sende de afviste asylansøgere hjem, der i dag bliver hængende uden lov til at være her. Det skal ske ved at love dem en bedre fremtid i hjemlandet, mener Venstre. DF og S kalder planen er urealistisk og ineffektiv.
Start of Article: Hjælp til at købe 100 kyllinger. Et bidrag til at købe en taxi. Eller måske støtte til at etablere en mekanikerbiks? EU bør have et fælles og fast finansieret økonomisk program, som kan lokke afviste asylansøgere uden lovligt ophold i EU til at rejse hjem og starte forfra. Det mener Venstres europaparlamentariker, Morten Løkkegaard, og udviklingsminister Ulla Tørnæs i et fælles udspil til en afrika- og migrationspolitik, hvor det, de kalder »hjemsendelsesstøtte,« er et centralt element. [...]

Figure 9: Abstractive summary that effectively summarises the salient information expressed through a long article.

Summary: Skuespilleren, som sprang fra rollen som Martin Rohde i dramaserien, følger ikke med i de nye afsnit
Start of Article: I de to første sæsoner af dramaserien 'Broen' havde svenske Saga Norén kollegialt seskab af danske Martin Rohde, spillet af Kim Bodnia. Men før tredje sæson sprang den danske skuespiller fra, da han var utilfreds med, hvordan hans rolle udviklede sig. I stedet er Thure Lindhardt blevet Sagas nye makker i DR1's hitserie. [...]

Figure 10: Abstractive summary which compresses the first five sentences into a single sentence.

Finally, Figure 11 shows another extractive summary where the entire summary may be found in the article. This time, it is as a single sentence, found as the second sentence in the article.

Summary: I Frankrig har en ulykke i forbindelse med et rallyløb kostet to personer livet, mens 15 personer blev kvæstet.
Start of Article: Ulykken skete i en lille by nær Toulon i Sydfrankrig. I Frankrig har en ulykke i forbindelse med et rallyløb kostet to personer livet, mens 15 personer blev kvæstet. Ifølge øjenvidner mistede føreren af bilen kontrollen i et sving. Bilen fortsatte med høj fart ind i en gruppe tilskuere. Den ene dræbte var en tilskuer, mens den anden var official ved løbet. Blandt de 15 kvæstede var mange børn Føreren af bilen slap med lettere kvæstelser. Ulykken skete i en lille by nær Toulon i Sydfrankrig. [...]

Figure 11: Extractive summary that uses entire sentence from article body as summary.

Paper III

MassiveSumm: a very large-scale, very multilingual, newswire summarisation dataset

Daniel Varab and Natalie Schluter
IT University of Copenhagen
Denmark
{djam,natschluter}@itu.dk

Abstract

Current research in automatic summarisation is unapologetically anglo-centered–a persistent state-of-affairs, which also predates neural net approaches. High-quality automatic summarisation datasets are notoriously expensive to create, posing a challenge for any language. However, with digitalisation, archiving, and social media advertising of newswire articles, recent work has shown how, with careful methodology application, large-scale datasets can now be simply gathered instead of written. In this paper, we present a large-scale multilingual summarisation dataset containing articles in 92 languages, spread across 28.8 million articles, in more than 35 writing scripts. This is both the largest, most inclusive, existing automatic summarisation dataset, as well as one of the largest, most inclusive, ever published datasets for any NLP task. We present the first investigation on the efficacy of resource building from news platforms in the low-resource language setting. Finally, we provide some first insight on how low-resource language settings impact state-of-the-art automatic summarisation system performance.

1 Introduction

Automatic summarisation datasets are generally expensive to create, because they generally involve a human reading a document several times and then crafting a fluent piece of text that captures both the important information of the document and the intention of the resulting summary. Each datapoint in such a dataset could take hours to manually create. With digitalisation, archiving, and social media advertising of newswire articles, recent work has shown how, with dedicated time and methodology application, large-scale datasets can now be simply gathered instead of written (Grusky et al., 2018; Hermann et al., 2015). But the method development was carried out over English, and until the research presented here, the method has only been applied to a very limited number of relatively richly-resourced languages (Varab and Schluter, 2020; Scialom et al., 2020).

We have extended the methodology further (Section 3) and applied it carefully and widely to generate MassiveSumm: a very large-scale, very multilingual summarisation dataset of 28.8 million articles, containing data in 92 languages, using more than 35 writing scripts. This is by far both the largest, most inclusive, existing automatic summarisation dataset, as well as one of the largest, most inclusive, ever published datasets for any NLP task. The bulk of this paper outlines the size, diversity and inclusivity of the dataset as an automatic summarisation dataset, as well as simply raw text data in comparison with two other multilingual large-scale widely used datasets in NLP: Wikipedia and Common Crawl (Section 4).

In light of extending and applying the data acquisition method under the low-resource setting, we identify some unreasonable conditions for language inclusion in automatic summarisation research, which stand to perpetuate a lack of language diversity in system development and therefore unequal access to these tools. We also present some experimental evidence that failure to include a more diverse set of language data in automatic summarisation research can result in only very language specific system design when language agnostic design has been claimed (Section 5).

2 Related Work

A number of works presenting large-scale datasets for automatic summarisation have been presented in the past couple of years. We survey this work here to provide some research context for MassiveSumm.

The New York Times Corpus (**NYT**) consists of 1.8 million articles from the New York Times (Sandhaus, 2008) between 1987 and 2007. The automatic summarisation portion of this dataset

consists of 650,000 article-summary pairs, where the summaries are written by library scientists. Unlike the rest of the datasets discussed in this section, NYT is created and maintained by the platform that the articles belong to.

The CNN/Daily Mail (**CNNDM**) dataset (Hermann et al., 2015) is an English language automatically acquired Question Answering dataset composed of newswire articles and their corresponding highlights from two separate platforms: cnn.com and dailymail.co.uk. The dataset was later converted into a summarisation dataset by concatenating these article highlights into article summaries (Cheng and Lapata, 2016; Nallapati et al., 2016). The summarisation dataset consists of 312,000 summary-article pairs. It has become the most broadly used automatically collected English summarisation dataset.

With the same methodology as CNNDM, Narayan et al. (2018) collected the **XSum** dataset of approximately 230,000 summary-article pairs from the bbc.com news platform. And Scialom et al. (2020) collected the **MLSum** dataset for five languages from five corresponding news platforms: French, German, Spanish, Russian, Turkish, catering their platform dependent method to each separate news platform. The resulting datset contains a total of around 1.5 million article-summary language pairs. MLSum was the first large-scale multilingual dataset, but all five of the languages of the dataset were still European, Indo-European, and relatively high-resourced within NLP. We note that while, similarly to XSum, MassiveSumm also contains article-summary pairs from the bbc.com platform, there are two important differences which make for zero overlap between the two datasets: (1) we include no English datapoints in our dataset, and (2) our summaries are not article highlights, but social media article descriptions, as is done for the remaining newswire datasets surveyed here.

The **Newsroom** dataset (Grusky et al., 2018) is the first large-scale English dataset generated specifically for automatic summarisation. The key insight into automatically creating this dataset was in observing use of a social media standard, called Open Graph[1], by publishers to improve their search engine results. According to this standard, a description of the article contents, used for advertising on social media, should be recorded in the mark-up of the article's web page. The method allowed for scraping news articles from any news outlet, so long as the news outlet upheld the social media standard. Hence, by contrast to the method for acquiring the CNNDM, Newsroom's method was website agnostic, which meant that scraping was no longer constrained to collecting data from specific platforms. Grusky et al. (2018) created Newsroom by conducting a scrape of news articles from 38 English language news outlets spanning two decades starting from the late 1990s, when news platforms first began digitalising their content widely, to 2017. The dataset contains 1.3 million document summary pairs.

Varab and Schluter (2020) extend, streamline and improve the Newsroom methodology to assemble the first automatic summarisation dataset for Danish, **DaNewsroom**. Their work comprises the first non-English website agnostic approach to large-scale article-summary collection, across 19 Danish news platforms and resulting in a dataset of 1.1M article-summary pairs. The methodology of this paper is adapted from this extension of the Newsroom methodology.

Related to this, the **GlobalVoices** dataset (Nguyen and Daumé III, 2019), is an automatic summarisation dataset across 15 languages from one single platform, https://globalvoices.org. Although its original collection is similar to Newsroom and DaNewsroom, the resulting dataset is relatively small with less than 30,000 article-summary pairs across all languages in total, including English. Moreover, approximately 800 English summaries are further crowdsourced. The dataset contains purely parallel data and its intended use is for cross-lingual summarisation. MassiveSumm most likely includes all non-English datapoints scraped for GlobalVoices, as this was one of the hundreds of its news platform data sources.

Two further large-scale datasets are not based on newswire. (1) **BigPatent** (Sharma et al., 2019) consists of 1.3 million U.S. patent English language abstract-document pairs, written between 1971 and 2018, across nine technological areas, all from the Google Patents Public Datasets (Google, 2018). (2) **LCSTS** The Large Scale Chinese Short Text Summarization Dataset (Hu et al., 2015) consists of 2.4 million text-summary pairs from the Sina Weibo microblogging platform, where post texts are paired with summaries provided by the author of each text.

[1] https://ogp.me/

Contemporaneously to our work, Hasan et al. (2021) developed **XL-Sum**, a summarisation dataset from the BBC news platform. However, their work covers less than a twelfth of the article-summary pairs: around 1 million across 44 languages and a single news platform, compared with our 12.3 million across 92 languages and 370 news platforms.

3 Methodology

Our methodology consists of roughly three parts: (1) manual annotation, (2) automatic collection, and (3) quality control. The first part is unique to the dataset presented here and represents a work-intensive annotation process which seeks to ensure both breadth in terms of language inclusivity, quality and consistency of the data. The remaining parts are measured adjustments of the prior extensions of Grusky et al. (2018)'s methodology by Varab and Schluter (2020).

Manual annotation. We first compiled a list of languages to be represented in the dataset. Our goal was to cover as many languages as possible, with a prioritisation of breadth, linguistic diversity, and language inclusivity, over depth. Then we manually searched for as many news platforms as possible for each language, by contrast to Grusky et al. (2018) who collected news platforms from publicly available lists.

For each news platform we required either (1) that it published exclusively in the language we had associated with it, or (2) published in way such that we could reliably distinguish the difference between languages later on (for example, the platform identified the languages for us). All other platforms were discarded.

Having determined which news platform were suitable language-wise, the next step was to manually investigate which platforms were technically *suitable*: we required these platforms to point to explicit lists of articles on their platform to avoid non-article content such as frontpages, albums or videos. In total, 370 different platforms met our requirements and were retained.

Automatic collection. With the list of suitable news platforms, we obtained all article URLs for each platform by retrieving them from `archive.org`. This is a slow process.

Having had collected the URLs for each platform we observe a significant difference between the amount of URLs across languages, some in the tens of millions, some in the thousands. We stored article URLs of the language together in language bins. We shuffled each bin and proceed to sample an equal amount of URLs from each bin and output them to a download queue. This allowed us to ensure that less frequent languages would always be scraped at the same priority as more frequent ones. Less frequent languages were sampled until they were exhausted, and thus over represented languages were sub-sampled.

Quality Control. We carry out a number of automatic checks for quality control, similarly to Varab and Schluter (2020). The number of articles filtered out of the dataset due to these checks can be seen in Table 1. In particular, we filter out articles with no contents, summaries with no contents, summaries that are prefixes of the article body, and summaries that are prefixes followed by "...". We quantify this filtering process in Section 4.

Distribution. Practically speaking, the publicly available dataset is distributed as a list of urls for each language (split into train/dev/test sets) and a single software package for downloading and processing the web pages.[2]

4 The numbers

Total counts. We refer to Table 1. Over 31 million articles were scraped from 370 news platforms, across 92 languages, from 38 language genera withing the following 16 language families: (1) Indo-European, (2) Afro-Asiatic, (3) Mande, (4) Niger-Congo, (5) Austronesian, (6) Altaic, (7) Sino-Tibetan, (8) Austro-Asiatic, (9) Kartvelian, (10) Uralic, (11) Japanese, (12) Dravidian, (13) Korean, (14) Tai-Kadai, (15) other, for Haïtian, and (16) Aymaran.[3] Of these, approximately 3 million scraped article pages had an empty article (2,981,925) and were filtered from the dataset, leaving over 28.8 million articles of raw multilingual text data, which we refer to as **MassiveSumm-All**.

As explained in Section 3, a number of filters were applied to the dataset to improve its quality for automatic summarisation. In particular, we did a check to ensure that summaries were neither empty nor just prefixes of the article, so that the

[2] https://github.com/danielvarab/massive-summ

[3] We took language family definitions and genus definitions from https://wals.info/ database.

	language	genus (family)	empty src	empty tgt	prefix	ellipsis	ellipsis|prefix	all-prefix	all-ellipsis	all	count	%invalid	valid count
						AFRICA							
2	Swahili	Bantoid (F4)	10,219	48,054	52,166	93,395	144,911	151,246	110,383	202,762	302,565	67.01%	99,803
3	Hausa	West Chadic (F13)	22,753	27,319	42,966	34,402	77,355	84,289	93,015	127,242	233,608	54.47%	106,366
4	Somali	Lowland East Cushitic (F13)	18,112	1,385	39,122	121,122	160,235	138,866	57,903	177,979	204,717	86.94%	26,738
5	Afrikaans	Germanic (F11)	374	8	121,056	5,549	126,173	5,927	121,434	126,551	198,792	63.66%	72,241
6	Kinyarwanda	Bantoid (F4)	17,791	6,878	40,893	21,241	62,062	45,307	65,477	86,128	92,674	92.94%	6,546
7	Amharic	Semitic (F13)	12,247	3,945	21,694	2,002	23,483	17,952	37,675	39,433	84,732	46.54%	45,299
8	North Ndebele	Bantoid (F4)	26,731	7	10,267	1,988	12,209	28,660	37,004	38,881	51,202	75.94%	12,321
9	Shona	Bantoid (F4)	25,130	5	12,505	715	13,205	25,840	37,638	38,330	46,681	82.11%	8,351
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
						EURASIA							
20	Russian	Slavic (F11)	26,564	27,482	432,521	91,252	491,426	145,096	486,458	545,270	1,284,433	42.45%	739,163
21	Spanish	Romance (F11)	36,434	101,844	85,805	428,547	513,726	564,728	223,487	649,907	1,216,217	53.44%	566,310
22	Ukrainian	Slavic (F11)	29,968	37,652	358,697	243,248	598,286	302,697	424,432	657,735	1,252,150	52.53%	594,415
23	Persian	Iranian (F11)	16,277	147,711	428,787	44,699	470,156	195,272	579,432	620,729	1,150,653	53.95%	529,924
24	Arabic	Semitic (F13)	44,039	216,084	403,561	6,296	408,247	263,573	661,071	665,524	1,186,870	56.07%	521,346
25	Chinese	Chinese (F9)	838,069	62,003	36,335	388,542	424,829	1,016,062	890,620	1,052,349	1,171,189	89.85%	118,840
26	German	Germanic (F11)	23,358	246,308	323,190	15,901	333,184	284,787	592,185	602,070	1,080,213	55.74%	478,143
27	Urdu	Indic (F11)	19,236	2,291	469,175	4,213	472,516	25,514	490,602	493,817	1,115,555	44.27%	621,738
28	Hindi	Indic (F11)	6,388	1,059	469,614	34,754	502,814	41,977	477,057	510,037	1,073,514	47.51%	563,477
29	French	Romance (F11)	31,711	112,622	249,625	323,869	564,598	458,696	388,211	699,425	1,007,129	69.45%	307,704
30	Polish	Slavic (F11)	6,808	39,910	435,591	22,334	454,093	68,471	482,246	500,230	983,252	50.88%	483,022
31	Vietnamese	Viet-Muong (F3)	532,441	21,410	125,609	81,298	199,344	590,681	672,481	708,727	920,166	77.02%	211,439
32	Bulgarian	Slavic (F11)	22,272	6,606	273,851	9,206	281,857	37,558	302,351	310,209	977,769	31.73%	667,560
33	Tamil	Southern Dravidian (F14)	1,074	11,654	703,881	126,331	829,332	138,242	715,826	841,243	886,482	94.90%	45,239
34	Hungarian	Ugric (F5)	17,332	28,724	220,577	1,229	221,511	43,082	262,478	263,364	885,749	29.73%	622,385
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
						INTERNATIONAL							
86	Esperanto	Constructed (F11)	0	0	27	103	130	103	27	130	565	23.01%	435
						NORTH AMERICA							
87	Haitian	Creoles and Pidgins (F6)	5,890	12	8,346	3,240	11,582	9,118	14,246	17,460	26,009	67.13%	8,549
						PAPUNESIA							
88	Indonesian	Malayo-Sumbawan (F12)	57,358	7,899	131,349	81,850	213,191	146,982	196,586	278,323	492,909	56.47%	214,586
89	Filipino	Greater Central Philippine (F12)	5	0	40	52	92	57	45	97	294	32.99%	197
90	Tetum	Central Malayo-Polynesian (F12)	0	0	2	0	2	0	2	2	15	13.33%	13
91	Bislama	Creoles and Pidgins (F6)	3	0	0	0	0	3	3	3	4	75.00%	1
						SOUTH AMERICA							
92	Aymara	Aymaran (F0)	32	0	110	104	213	129	142	238	827	28.78%	589
		totals	2,981,925	1,775,581	10,315,099	5,145,760	15,238,148	9,404,789	14,891,856	19,497,177	31,940,180	58.04%	12,443,003

LANGUAGE FAMILY LEGEND

- Aymaran (F0)
- Kartvelian (F1)
- Altaic (F2)
- Austro-Asiatic (F3)
- Niger-Congo (F4)
- Uralic (F5)
- other (F6)
- Japanese (F7)
- Tai-Kadai (F8)
- Sino-Tibetan (F9)
- Mande (F10)
- Indo-European (F11)
- Austronesian (F12)
- Afro-Asiatic (F13)
- Dravidian (F14)

Table 1: Excerpt language article-summary pair counts from Table 8 in the Appendix. In the table columns, **empty_art** is the number of articles with no contents, **empty_sum** is the number of summaries with not contents, **prefix** is the number of summaries that also prefixes of the article, **ellipsis** is the number of summaries that are prefixes of the article followed by "...", **ellipsis|prefix** is the number of either ellipsis or prefix summaries (they are not mutually exclusive), **all-prefix** is the number of summaries after filtering, but including prefixes, **all-ellipsis** is the number of summaries after filtering, but including ellipsis, **all** is the number of empty, prefix or ellipsis summaries (they are not mutually exclusive), **count** is the total number of article-summary pairs, **%invalid** is the proportion of filtered article-summary pairs (all/count), and **valid count** is the number of article-summary pairs after filtering.

resulting dataset did not include trivial instances for system development. MassiveSumm can therefore be seen under two views: **MassiveSumm-All (MS-All)** which consists of all non-empty articles (and any available summaries) before application of the above-mentioned filters. And a subset of this– the **MassiveSumm (MS)** summarisation dataset intended for automatic summarisation system development; this dataset is the result of the application of the filters.

We observe (Table 2) that the majority of the dataset, approximately 16.5 million article-summary pairs, did not survive the summary quality control filtering process. The result was 12,368,113 article-summary pairs surviving a minimal quality control for utility in automatic summarisation system development, of which the automatic-summarisation dataset portion of MassiveSumm consists.

dataset	description	size
MassiveSumm (MS)	Fully filtered automatic summarisation data.	12,368,113 article-summary pairs.
MassiveSumm-All (MS-All)	All non-empty articles scraped.	28,879,290 articles.

Table 2: Summary of the contents of MassiveSumm.

This filtering process resulted in a handful of languages having virtually no presence in the automatic summarisation portion of MassiveSumm. For instance, over 98.7% of Xhosa article-summary pairs were filtered out of the summarisation portion of the dataset, leaving only 172 instances.

Table 3 gives an overview of the article/article-summary pair counts. We note that the Indo-European languages provide the majority of the data in the dataset. The Uralic family (here, only with Hungarian) is also relatively heavily represented in the dataset. The 10 Niger-Congo languages as a whole have less data than a single Indo-European language on average. In Section 5 we discuss why our current methodology can only result in perpetuating such under-representation in dataset quantities.

Comparing with web-scrape multilingual datasets. We compared the intersection of our dataset with two large-scale web datasets widely used by the NLP community: Wikipedia[4] and Common Crawl[5]. An overview of this comparison can be found in Table 4. The manual care that we took in curating the list of platforms from which we wanted to collect data resulted in more data from an improved diversity of languages.

For 52 of our languages, MS-All either matches or surpasses the number of Wikipedia pages for the language in question, showing the importance of the full dataset simply as raw data. In fact, the majority of MassiveSumm languages from South Saharan Africa (14/18) have more documents in MS-All than in Wikipedia. And well over half of the MassiveSumm languages for Eurasia (38/63) have more documents in MS-All than in Wikipedia.

Turning to Common Crawl, almost half of the languages from South Saharan Africa (8/18) have more pages in MS-All than in Common Crawl. Six out of 63 Eurasian languages have more articles in MS-All than in Common Crawl.

When we consider even just the heavily filtered automatic summarisation portion of the data, MS, we find that 10 of the South Saharan African languages contain more pages than Wikipedia, and 5 out of 18 of these languages contain more data than Common Crawl. For Eurasia, 19 of the 63 languages contain more pages than Wikipedia.

Table 5 gives the proportions of the articles in MS-All that are also contained in Common Crawl, for those languages where more than 49% can be obtained. This is 18 languages–around a fifth of the languages represented by MassiveSumm. Hence observe that large portions of easily indexible and crawlable, publicly available, diverse linguistic data are not being scraped into one of the most important datasets for NLP, both in size, but in determining to a large extent which languages get mainstream NLP research: Common Crawl.

5 Reflections on Low-Resource Language Automatic Summarisation

The central datasets for automatic summarisation have consistently been for English. In this section we consider how this focus on English has resulted in limited dataset curation methodology development (Section 5.1) and limited automatic summarisation system design (Section 5.2).

[4] https://en.wikipedia.org/wiki/List_of_Wikipedias#Edition_detailsasofMay10, 2021

[5] April 2021 crawl CC-MAIN-2021-04 https://commoncrawl.github.io/cc-crawl-statistics/plots/languages.csv

family	MS-All	%(MS-All)	MS	%(MS)	num langs	MS-All ave	MS-All ave%	MS ave	MS ave%
Indo-European	20990245	72.68%	9062565	73.27%	48	437296.77	1.51%	188803.44	1.53%
Dravidian	2005933	6.95%	333765	2.7%	4	501483.25	1.74%	83441.25	0.67%
Afro-Asiatic	1753871	6.07%	816504	6.6%	7	250553.0	0.87%	116643.43	0.94%
Uralic	868417	3.01%	622385	5.03%	1	868417.0	3.01%	622385.0	5.03%
Altaic	835649	2.89%	362191	2.93%	5	167129.8	0.58%	72438.2	0.59%
Austro-Asiatic	477331	1.65%	257197	2.08%	2	238665.5	0.83%	128598.5	1.04%
Niger-Congo	467630	1.62%	142921	1.16%	10	46763.0	0.16%	14292.1	0.12%
Austronesian	462877	1.6%	232510	1.88%	4	115719.25	0.4%	58127.5	0.47%
Sino-Tibetan	434543	1.5%	177373	1.43%	3	144847.67	0.5%	59124.33	0.48%
Tai-Kadai	252073	0.87%	132287	1.07%	2	126036.5	0.44%	66143.5	0.53%
Kartvelian	182743	0.63%	132055	1.07%	1	182743.0	0.63%	132055.0	1.07%
Japanese	125625	0.44%	87220	0.71%	1	125625.0	0.44%	87220.0	0.71%
other	20120	0.07%	8550	0.07%	2	10060.0	0.03%	4275.0	0.03%
Mande	1438	0.0%	1	0.0%	1	1438.0	0.0%	1.0	0.0%
Aymaran	795	0.0%	589	0.0%	1	795.0	0.0%	589.0	0.0%
Totals	28879290		12368113		92				

Table 3: Language family-wise article counts and proportions for MassiveSumm-All (All) and for the MassiveSumm automatic summarisation dataset (MS).

5.1 Impact on dataset curation

The methodology we use for acquiring this dataset is based on Newsroom (Grusky et al., 2018), a dataset for English. In order for the method to be effective at obtaining data, at least the following two assumptions must be met.

Assumption 1. Digitalisation. Digitised newswire text must be publicly available online for the language, and in sufficiently large quantities. This is not the case, however. For example, a broad manual search for online news platforms in Africa[6] revealed relatively few non-colonial language platforms for the region. Digitised newswire is also sparse or non-existent in, for example, non-standard Arabic dialects, European languages such as Irish or Welsh, as well as indigenous languages in North and South America, and Australia. Hence focus on a strategy created for a language where there are massive amounts of online data, and lack of development of new techniques to acquire data for languages that do not have such an online presence will reinforce the lack of representation of these languages in automatic summarisation research.

Assumption 2. Web page structure conventions. Online news platforms must ensure that their article mark-ups abide by the Open Graph protocol (Cf. Section 3). However, extensive manual inspection revealed that while this is the norm for English and in general for languages of rich western countries, this is not the norm in general. For instance, due to this problem we had to exclude a number of other South Saharan African languages including Southern Sotho, Pulaar, Zulu, and Luganda. Further, as we observe in Table 1, approximately 2 million documents are excluded from MS due to their summaries being empty–the news platforms in the corresponding languages have the correct template structure for their web pages, but do not use them as intended.

In order to develop the know-how to achieve true language diversity in datasets for automatic summarisation (and other NLP tasks), methods for acquiring automatic summarisation data should be developed which do not make these two assumptions. The difference in existence and in quantities of data for the languages of MassiveSumm reflect this requirement, which currently favours Indo-European languages.

5.2 Systems: Low-resource baselines

MassiveSumm provides a means to check whether there is evidence of some impact of a focus on English data for neural automatic summarisation.

The languages. We consider a minimal set of non-Indo-European languages to provide such evidence according to three separate considerations: (1) The languages should have large native speaker populations.[7]. (2) The languages should be non-Indo-European. (3) The set of languages should exhibit different complexity in morphology. (4) The datasets should be of significantly different sizes. (5) Finally, all languages must have readily available word segmentisers.

The set of languages we chose for our experiments all have a population far beyond that of the

[6] https://www.w3newspapers.com/africa/

[7] According to https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers

language	family	MS	MS-all	Wiki	CC	MS/Wiki	MS/CC	MS-All/Wiki	MS-All/CC
				AFRICA					
Amharic	Afro-Asiatic	45,299	72,485	14,910	95,305	**303.82%**	47.53%	**486.15%**	76.06%
Bambara	Mande	1	1,438	693	0	0.14%	-	**207.50%**	-
Fulah	Niger-Congo	40	499	278	0	14.39%	-	**179.50%**	-
Hausa	Afro-Asiatic	106,366	210,855	6,829	54,355	**1557.56%**	**195.69%**	**3087.64%**	**387.92%**
Igbo	Niger-Congo	4,341	5,085	2,084	9,728	**208.30%**	44.62%	**244.00%**	52.27%
Lingala	Niger-Congo	1,489	4,429	3,184	5,224	46.77%	28.50%	**139.10%**	84.78%
North Ndebele	Niger-Congo	12,321	24,471	0	0	-	-	-	-
Oromo	Afro-Asiatic	5,816	14,926	1,050	15,432	**553.90%**	37.69%	**1421.52%**	96.72%
Rundi	Niger-Congo	3,646	24,085	618	2,882	**589.97%**	**126.51%**	**3897.25%**	**835.70%**
Shona	Niger-Congo	8,351	21,551	6,660	11,559	**125.39%**	72.25%	**323.59%**	**186.44%**
Somali	Afro-Asiatic	26,738	186,605	5,944	159,270	**449.83%**	16.79%	**3139.38%**	**117.16%**
Swahili	Niger-Congo	99,803	292,346	60,725	243,070	**164.35%**	41.06%	**481.43%**	**120.27%**
Tigrinya	Afro-Asiatic	7,978	18,533	208	25,369	**3835.58%**	31.45%	**8910.10%**	73.05%
Xhosa	Niger-Congo	172	12,876	1,182	37,430	14.55%	0.46%	**1089.34%**	34.40%
				EURASIA					
Albanian	Indo-European	156,336	680,535	82,309	1,296,319	**189.94%**	12.06%	**826.81%**	52.50%
Arabic	Afro-Asiatic	521,346	1,142,831	1,102,405	19,101,195	47.29%	2.73%	**103.67%**	5.98%
Armenian	Indo-European	168,453	807,817	281,101	1,050,372	59.93%	16.04%	**287.38%**	76.91%
Azerbaijani	Altaic	140,685	301,134	177,955	1,548,046	79.06%	9.09%	**169.22%**	19.45%
Bengali	Indo-European	124,351	191,712	103,686	2,681,993	**119.93%**	4.64%	**184.90%**	7.15%
Bosnian	Indo-European	45,575	254,737	84,968	1,311,659	53.64%	3.47%	**299.80%**	19.42%
Bulgarian	Indo-European	667,560	955,497	269,103	9,070,911	**248.07%**	7.36%	**355.07%**	10.53%
Central Khmer	Austro-Asiatic	45,758	89,606	8,230	300,772	**555.99%**	15.21%	**1088.77%**	29.79%
Czech	Indo-European	551,443	609,257	473,960	36,586,487	**116.35%**	1.51%	**128.55%**	1.67%
Dari	Indo-European	20,220	59,199	0	0	-	-	-	-
Georgian	Kartvelian	132,055	182,743	148,069	1,269,380	89.18%	10.40%	**123.42%**	14.40%
Gujarati	Indo-European	43,830	450,740	29,481	294,393	**148.67%**	14.89%	**1528.92%**	**153.11%**
Hindi	Indo-European	563,477	1,067,126	145,723	4,185,074	**386.68%**	13.46%	**732.30%**	25.50%
Hungarian	Uralic	622,385	868,417	483,555	18,592,776	**128.71%**	3.35%	**179.59%**	4.67%
Kannada	Dravidian	47,676	281,630	26,789	309,943	**177.97%**	15.38%	**1051.29%**	90.87%
Kurdish	Indo-European	28,008	94,916	37,232	204,372	75.23%	13.70%	**254.93%**	46.44%
Lao	Tai-Kadai	40,316	53,193	3,594	103,238	**1121.76%**	39.05%	**1480.05%**	51.52%
Latvian	Indo-European	7,080	454,915	105,928	2,970,478	6.68%	0.24%	**429.46%**	15.31%
Lithuanian	Indo-European	326,082	884,547	201,003	5,362,226	**162.23%**	6.08%	**440.07%**	16.50%
Macedonian	Indo-European	86,647	219,869	112,077	889,870	77.31%	9.74%	**196.18%**	24.71%
Malayalam	Dravidian	121,568	634,601	71,996	676,894	**168.85%**	17.96%	**881.44%**	93.75%
Marathi	Indo-European	127,838	476,870	69,262	496,649	**184.57%**	25.74%	**688.50%**	96.02%
Modern Greek	Indo-European	95,023	401,315	188,407	18,299,263	50.43%	0.52%	**213.00%**	2.19%
Nepali	Indo-European	23,993	218,138	31,745	805,140	75.58%	2.98%	**687.16%**	27.09%
Oriya	Indo-European	28,582	388,961	15,592	122,957	**183.31%**	23.25%	**2494.62%**	**316.34%**
Panjabi	Indo-European	83,147	322,520	35,218	168,347	**236.09%**	49.39%	**915.78%**	**191.58%**
Persian	Indo-European	529,924	1,134,376	767,776	20,893,043	69.02%	2.54%	**147.75%**	5.43%
Pushto	Indo-European	58,038	215,927	11,807	90,702	**491.56%**	63.99%	**1828.80%**	**238.06%**
Scottish Gaelic	Indo-European	15,012	16,528	15,198	48,315	98.78%	31.07%	**108.75%**	34.21%
Sinhala	Indo-European	12,252	32,851	16,818	215,962	72.85%	5.67%	**195.33%**	15.21%
Slovak	Indo-European	78,639	581,873	235,863	12,240,989	33.34%	0.64%	**246.70%**	4.75%
Tamil	Dravidian	45,239	885,408	134,646	1,444,153	33.60%	3.13%	**657.58%**	61.31%
Telugu	Dravidian	119,282	204,294	70,641	573,248	**168.86%**	20.81%	**289.20%**	35.64%
Thai	Tai-Kadai	91,971	198,880	142,059	11,108,049	64.74%	0.83%	**140.00%**	1.79%
Tibetan	Sino-Tibetan	1,236	6,455	5,949	32,107	20.78%	3.85%	**108.51%**	20.10%
Ukrainian	Indo-European	594,415	1,222,182	1,073,297	12,688,368	55.38%	4.68%	**113.87%**	9.63%
Urdu	Indo-European	621,738	1,096,319	160,631	725,101	**387.06%**	85.75%	**682.51%**	**151.20%**
Welsh	Indo-European	53,802	154,844	132,464	358,792	40.62%	15.00%	**116.90%**	43.16%

Table 4: Languages for which MassiveSumm carries more raw documents than Wikipedia or Common Crawl.

average European country. And yet two of these languages are severely lower resourced in NLP in general, if not zero-resourced. The languages are:

- **Arabic**, a semitic language with a complex morphology and around 310 million native speakers. We used 432,384 article-summary pairs from MS.
- **Telugu**, a Dravidian language with a moderately rich morphology and around 82 million native speakers. We used 12,633 article-summary pairs from MS.
- **Hausa**, an Afro-Asiatic tonal language with a relatively simple morphology and around 43 million native speakers. We used 78,633 article-summary pairs from MS.

The datasets were split into train/test/dev sets with corresponding proportions 80%/10%/10%. For tokenisation of Arabic and Telugu we used Spacy (Honnibal et al., 2020), and the English tokeniser from NLTK (Loper and Bird, 2002) for Hausa. For sentence segmentation we use `pySBD` (Sadvilkar and Neumann, 2020) for Arabic, and NLTK for the remaining Hausa and Telugu.

The system. OpenNMT's (Klein et al., 2017) reimplementation of the Pointer-Generator system (See et al., 2017) provides efficient state-of-the-

language	%	language	%
Tibetan	96.98%	Lingala	72.05%
Lao	95.45%	Malagasy	67.18%
Bambara	95.37%	Tigrinya	66.69%
Dari	94.87%	Bosnian	63.71%
Rundi	84.28%	Scot. Gaelic	63.71%
Burmese	81.51%	Hungarian	61.30%
Haitian	79.50%	Slovenian	58.21%
Oromo	77.93%	Bislama	50.00%
Kurdish	74.77%	Irish	49.27%

Table 5: Languages from MassiveSumm-All for which the percentage of articles that can also be found in Common Crawl is greater than 49%.

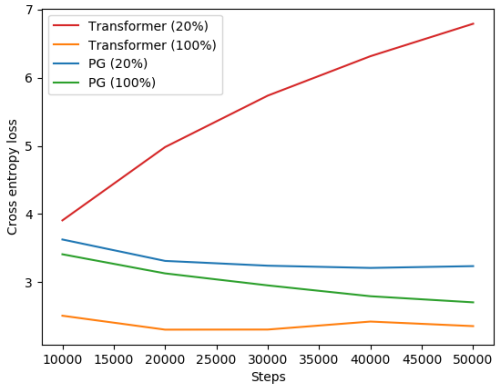


Figure 1: Two fixed architecture configurations run under two data settings: (1) 100% of the training set, and (2) 20% of the training set. The PG model (rnn) is robust to different data settings while the transformer quickly overfits the training data. Loss in the graph is measured over the development set.

art-competitive performance and proved more robust to limits in dataset size than a Transformer (Vaswani et al., 2017) model during our hyperparameter search preparatory experiments–this was a crucial requirement for our low-resource language experiments. We experimented with training both Pointer-Generator and Transformer models over different quantities, 20% and 100% (respectively, 57,444 and 287,227 instances), of CNNDM training data. While the transformer outperforms PG when training on the full dataset (Table 6), it grossly overfits when faced with only 20% of the data for training (Figure 1).

system (train prop.)	R1	R2	RL
Transformer (100% data)	39.06	17.02	36.09
Transformer (20% data)	32.23	11.12	29.99
PG (100%)	38.41	16.31	35.21
PG (80%)	38.18	16.3	35.08
PG (60%)	38.13	16.16	34.92
PG (40%)	38.05	16.13	34.9
PG (20%)	36.81	15.36	33.7

Table 6: Rouge-1, Rouge-2, and Rouge-L (Lin, 2004) scores for comparing Transformer and RNN (PG) models on different proportions of CNNDM training data in preparation for lower-resource language experiments.

For further context, we also train and test on the Newsroom corpus. Since the Newsroom corpus did not filter prefix and ellipsis summaries, we include scores with and without these data filters. We use an 80%/10%/10% split of Newsroom before and after filtering: respectively 994,446/109,147/109,147 and 808,727/88,657/88,768 article-summary pairs.

During training we truncate articles to 400 tokens and summaries to 100 tokens. We fix the random seed but refrain from tying the input and output embeddings (Press and Wolf, 2016). The vocabularies are fixed to 30,000 tokens across all languages and we used no subword tokeniser. At inference time we decoded with a beam size of 10, discarded summaries with less than 35 tokens, block trigrams and apply length penalty with the value $\alpha = 0.9$ (Wu et al., 2016). For further details of the model, we refer to the original papers of (See et al., 2017; Gehrmann et al., 2018) as well as OpenNMT's documentation[8]. Our experiments should act as lower bounds as we conducted no tuning on any of the MassiveSumm datasets.

We include the Lead-3 baseline which simply copies the first three sentences from the article. It is a notoriously strong baseline for automatic summarisation systems and acts as a baseline point of reference that is resilient to training set size limitations.

The results are given in Table 7. In particular, we notice that ROUGE scores tend to be rather low for the largest non-English dataset, Arabic, with the most complex morphology, despite being the largest of the three. As expected, Telugu with the smallest dataset, also has low ROUGE scores. On the other hand, Lead-3 performs better but similarly low in ROUGE score. On the other hand, ROUGE scores for Hausa are significantly higher in scale than Newsroom scores and also significantly outperform the strong Lead-3 baseline. We have 3

[8] https://opennmt.net/OpenNMT-py/examples/Summarization.html

different linguistic contexts and three quite different behaviours, which provides clear evidence that robust development in automatic summarisation must adjust and consider linguistic diversity.

dataset (system)	R1	R2	RL
Arabic (Point.-Gen.)	13.58	4.02	13.53
Arabic (Lead-3)	11.34	3.18	11.27
Hausa (Point.-Gen.)	38.55	28.5	31.91
Hausa (Lead-3)	30.55	17.95	26.68
Telugu (Point.-Gen.)	5.62	1.43	5.62
Telugu (Lead-3)	8.87	2.17	8.7
Newsroom (Point.-Gen.)	34.73	21.25	30.39
Newsroom (Lead-3)	31.12	21.4	28.49
Filt. Newsroom (Point.-Gen.)	28.95	15.5	23.9
Filt. Newsroom (Lead-3)	25.49	14.17	22.49

Table 7: Baseline ROUGE scores for Arabic, Hausa, and Telugu. ROUGE scores for Newsroom added for context.

6 Concluding Remarks

In this paper, we presented the most large-scale, most language and linguistically diverse and inclusive dataset for automatic summarisation to date: MassiveSumm. In acquiring MassiveSumm, we also acquired one of the most diverse and inclusive sources of raw linguistic data to date. We also provided evidence how focus on anglo-centric data acquisition method development and system development were detrimental to both language inclusion and language agnostic system behaviour.

References

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494, Berlin, Germany. Association for Computational Linguistics.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, Brussels, Belgium. Association for Computational Linguistics.

Google. 2018. Google patents public datasets: connecting public, paid, and private patent data.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. 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 708–719, New Orleans, Louisiana. Association for Computational Linguistics.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703, Online. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28*, pages 1693–1701. Curran Associates, Inc.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Baotian Hu, Qingcai Chen, and Fangze Zhu. 2015. LCSTS: A large scale Chinese short text summarization dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1967–1972, Lisbon, Portugal. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018*

Conference on Empirical Methods in Natural Language Processing, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Khanh Nguyen and Hal Daumé III. 2019. Global Voices: Crossing borders in automatic news summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 90–97, Hong Kong, China. Association for Computational Linguistics.

Ofir Press and Lior Wolf. 2016. Using the output embedding to improve language models. *arXiv preprint arXiv:1608.05859*.

Nipun Sadvilkar and Mark Neumann. 2020. PySBD: Pragmatic sentence boundary disambiguation. In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, pages 110–114, Online. Association for Computational Linguistics.

Evan Sandhaus. 2008. The new york times annotated corpus. Technical report, Linguistic Data Consortium, Philadelphia.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2020. MLSUM: The multilingual summarization corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8051–8067, Online. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Eva Sharma, Chen Li, and Lu Wang. 2019. BIGPATENT: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2204–2213, Florence, Italy. Association for Computational Linguistics.

Daniel Varab and Natalie Schluter. 2020. DaNewsroom: A large-scale Danish summarisation dataset. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6731–6739, Marseille, France. European Language Resources Association.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

A Appendix

This appendix contains the full version of Table 1.

	language	genus (family)	empty src	empty tgt	prefix	ellipsis	ellipsis|prefix	all-prefix	all-ellipsis	all	count	%invalid	valid count
						AFRICA							
1	Swahili	Bantoid (F4)	10,219	48,054	52,166	93,395	144,911	151,246	110,383	202,762	302,565	67.01%	99,803
2	Hausa	West Chadic (F13)	22,753	27,319	42,966	34,402	77,355	84,289	93,015	127,242	233,608	54.47%	106,366
3	Somali	Lowland East Cushitic (F13)	18,112	1,385	39,122	121,122	160,235	138,866	57,903	177,979	204,717	86.94%	26,738
4	Afrikaans	Germanic (F11)	374	8	121,056	5,549	126,173	5,927	121,434	126,551	198,792	63.66%	72,241
5	Kinyarwanda	Bantoid (F4)	17,791	6,878	40,893	21,241	62,062	45,307	65,477	86,128	92,674	92.94%	6,546
6	Amharic	Semitic (F13)	12,247	3,945	21,694	2,002	23,483	17,952	37,675	39,433	84,732	46.54%	45,299
7	North Ndebele	Bantoid (F4)	26,731	7	10,267	1,988	12,209	28,660	37,004	38,881	51,202	75.94%	12,321
8	Shona	Bantoid (F4)	25,130	5	12,505	715	13,205	25,840	37,638	38,330	46,681	82.11%	8,351
9	Rundi	Bantoid (F4)	7,478	74	8,576	11,840	20,416	19,341	16,126	27,917	31,563	88.45%	3,646
10	Tigrinya	Semitic (F13)	6,625	77	4,261	6,262	10,522	12,920	10,957	17,180	25,158	68.29%	7,978
11	Oromo	Lowland East Cushitic (F13)	7,315	14	1,811	7,325	9,135	14,615	9,139	16,425	22,241	73.85%	5,816
12	Malagasy	Barito (F12)	23	0	3,741	5,603	9,318	5,616	3,764	9,331	27,045	34.50%	17,714
13	Xhosa	Bantoid (F4)	124	2	236	12,483	12,718	12,593	360	12,828	13,000	98.68%	172
14	Bambara	Western Mande (F10)	6,137	1	1,437	14	1,451	6,137	7,574	7,574	7,575	99.99%	1
15	Yoruba	Defoid (F4)	33	10	639	555	1,193	588	676	1,226	7,438	16.48%	6,212
16	Igbo	Igboid (F4)	8	492	171	83	253	582	670	752	5,093	14.77%	4,341
17	Lingala	Bantoid (F4)	305	0	2,938	5	2,943	307	3,243	3,245	4,734	68.55%	1,489
18	Fulah	Peul-Serer (F4)	0	215	38	207	244	422	253	459	499	91.98%	40
						EURASIA							
19	Russian	Slavic (F11)	26,564	27,482	432,521	91,252	491,426	145,096	486,458	545,270	1,284,433	42.45%	739,163
20	Spanish	Romance (F11)	36,434	101,844	85,805	428,547	513,726	564,728	223,487	649,907	1,216,217	53.44%	566,310
21	Ukrainian	Slavic (F11)	29,968	37,652	358,697	243,248	598,286	302,697	424,432	657,735	1,252,150	52.53%	594,415
22	Persian	Iranian (F11)	16,277	147,711	428,787	44,699	470,156	195,272	579,432	620,729	1,150,653	53.95%	529,924
23	Arabic	Semitic (F13)	44,039	216,084	403,561	6,296	408,247	263,573	661,071	665,524	1,186,870	56.07%	521,346
24	Chinese	Chinese (F9)	838,069	62,003	36,335	388,542	424,829	1,016,062	890,620	1,052,349	1,171,189	89.85%	118,840
25	German	Germanic (F11)	23,358	246,308	323,190	15,901	333,184	284,787	592,185	602,070	1,080,213	55.74%	478,143
26	Urdu	Indic (F11)	19,236	2,291	469,175	4,213	472,516	25,514	490,602	493,817	1,115,555	44.27%	621,738
27	Hindi	Indic (F11)	6,388	1,059	469,614	34,754	502,814	41,977	477,057	510,037	1,073,514	47.51%	563,477
28	French	Romance (F11)	31,711	112,622	249,625	323,869	564,598	458,696	388,211	699,425	1,007,129	69.45%	307,704
29	Polish	Slavic (F11)	6,808	39,910	435,591	22,334	454,093	68,471	482,246	500,230	983,252	50.88%	483,022
30	Vietnamese	Viet-Muong (F3)	532,441	21,410	125,609	81,298	199,344	590,681	672,481	708,727	920,166	77.02%	211,439
31	Bulgarian	Slavic (F11)	22,272	6,606	273,851	9,206	281,857	37,558	302,351	310,209	977,769	31.73%	667,560
32	Tamil	Southern Dravidian (F14)	1,074	11,654	703,881	126,331	829,332	138,242	715,826	841,243	886,482	94.90%	45,239
33	Hungarian	Ugric (F5)	17,332	28,724	220,577	1,229	221,511	43,082	262,478	263,364	885,749	29.73%	622,385
34	Lithuanian	Baltic (F11)	335	100,060	131,465	327,472	458,586	427,686	231,826	558,800	884,882	63.15%	326,082
35	Armenian	Armenian (F11)	15,906	15,450	107,732	531,897	639,295	547,872	124,117	655,270	823,723	79.55%	168,453
36	Kannada	Southern Dravidian (F14)	502,488	5,767	205,491	44,631	246,047	535,026	711,609	736,442	784,118	93.92%	47,676
37	Italian	Romance (F11)	3,172	227,502	20,405	26,676	46,996	256,974	250,819	277,294	885,915	31.30%	608,621
38	Albanian	Albanian (F11)	4,524	9,133	509,787	6,381	515,192	19,912	523,416	528,723	685,059	77.18%	156,336
39	Malayalam	Southern Dravidian (F14)	4,125	169	118,459	395,556	513,530	399,184	122,743	517,158	638,726	80.97%	121,568
40	Czech	Slavic (F11)	148	1,010	52,020	5,143	56,826	6,279	53,156	57,962	609,405	9.51%	551,443
41	Slovak	Slavic (F11)	668	25,599	477,616	118,930	477,946	144,886	503,576	503,902	582,541	86.50%	78,639
42	Marathi	Indic (F11)	925	7,158	332,233	9,749	341,935	17,771	340,308	349,957	477,795	73.24%	127,838
43	Gujarati	Indic (F11)	422	2,035	6,455	398,661	405,103	400,890	8,882	407,332	451,162	90.29%	43,830
44	Oriya	Indic (F11)	37,874	36,075	177,446	180,032	357,429	220,856	219,146	398,253	426,835	93.30%	28,582
45	Modern Greek	Greek (F11)	4,755	10,094	232,570	69,587	296,865	83,769	246,787	311,047	406,070	76.60%	95,023
46	Turkish	Turkic (F2)	5,172	2,453	254,896	5,308	257,354	11,759	261,427	263,805	376,891	70.00%	113,086
47	Portuguese	Romance (F11)	21,362	36,428	72,366	107,697	179,571	165,336	130,131	237,210	374,602	63.32%	137,392
48	Latvian	Baltic (F11)	7,950	10,535	444,534	2,746	444,752	13,779	455,567	455,785	462,865	98.47%	7,080

language family
- Aymaran (F0)
- Kartvelian (F1)
- Altaic (F2)
- Austro-Asiatic (F3)
- Niger-Congo (F4)
- Uralic (F5)
- other (F6)
- Japanese (F7)
- Tai-Kadai (F8)
- Sino-Tibetan (F9)
- Mande (F10)
- Indo-European (F11)
- Austronesian (F12)
- Afro-Asiatic (F13)
- Dravidian (F14)

Table 8: Full table of language article-summary pair counts. In the table columns, **empty_art** is the number of articles with no contents, **empty_sum** is the number of summaries with not contents, **prefix** is the number of summaries that also prefixes of the article, **ellipsis** is the number of summaries that are prefixes of the article followed by "...", **ellipsis|prefix** is the number of either ellipsis or prefix summaries (they are not mutually exclusive), **all-prefix** is the number of summaries after filtering, but including prefixes, **all-ellipsis** is the number of summaries after filtering, but including ellipsis, **all** is the number of empty, prefix or ellipsis summaries (they are not mutually exclusive), **count** is the total number of article-summary pairs, **%invalid** is the proportion of filtered article-summary pairs (all/count), and **valid count** is the number of article-summary pairs after filtering. (Table continued on next page.)

	language	genus (family)	empty src	empty tgt	prefix	ellipsis	ellipsis|prefix	all-prefix	all-ellipsis	all	count	%invalid	valid count
						EURASIA CONT'D							
49	Azerbaijani	Turkic (F2)	3,910	737	143,471	16,894	159,757	21,496	148,099	164,359	305,044	53.88%	140,685
50	Bosnian	Slavic (F11)	7,284	15,627	92,486	107,925	200,120	124,251	108,842	216,446	262,021	82.61%	45,575
51	Pushto	Iranian (F11)	45,882	27,965	106,804	48,889	155,642	97,018	154,958	203,771	261,809	77.83%	58,038
52	Thai	Kam-Tai (F8)	23,309	13,098	41,198	59,015	96,445	92,788	75,715	130,218	222,189	58.61%	91,971
53	Nepali	Indic (F11)	725	23,181	58,859	112,622	171,476	136,016	82,670	194,870	218,863	89.04%	23,993
54	Macedonian	Slavic (F11)	449	368	87,230	45,901	133,010	46,562	87,947	133,671	220,318	60.67%	86,647
55	Panjabi	Indic (F11)	6,353	2,471	218,108	19,491	237,043	28,174	226,916	245,726	328,873	74.72%	83,147
56	Icelandic	Germanic (F11)	2,167	15	95,095	63,965	158,994	66,081	97,276	161,110	199,970	80.57%	38,860
57	Bengali	Indic (F11)	32,106	940	50,535	16,175	66,577	49,065	83,544	99,467	223,818	44.44%	124,351
58	Japanese	Japanese (F7)	74,711	139	38,179	8,515	46,694	74,937	112,893	113,116	200,336	56.46%	87,220
59	Telugu	South-Central Dravidian (F14)	5,421	2,845	74,573	11,179	83,273	18,339	82,030	90,433	209,715	43.12%	119,282
60	English	Germanic (F11)	184	65	161,571	1,463	162,062	1,696	161,808	162,295	209,021	77.65%	46,726
61	Georgian	Kartvelian (F1)	4,097	2,785	32,106	16,185	48,175	22,795	38,765	54,785	186,840	29.32%	132,055
62	Slovenian	Slavic (F11)	2,269	17	20,113	2,702	22,421	4,977	22,397	24,696	168,688	14.64%	143,992
63	Burmese	Burmese-Lolo (F9)	65,254	190	35,508	7,579	43,045	67,459	100,803	102,925	160,222	64.24%	57,297
64	Welsh	Celtic (F11)	1,915	33	100,617	2,197	101,036	4,118	102,565	102,957	156,759	65.68%	53,802
65	Tajik	Iranian (F11)	501	13	86,260	2,323	88,518	2,837	86,774	89,032	150,419	59.19%	61,387
66	Kurdish	Iranian (F11)	52,706	652	58,405	7,908	66,292	61,230	111,747	119,614	147,622	81.03%	28,008
67	Serbian	Slavic (F11)	6,554	16,395	15,784	38,628	54,248	60,395	38,693	76,015	189,144	40.19%	113,129
68	Uzbek	Turkic (F2)	8,100	376	41,983	3,858	45,544	11,870	50,216	53,556	138,748	38.60%	85,192
69	Hebrew	Semitic (F13)	6	4,011	41	623	664	4,640	4,058	4,681	107,642	4.35%	102,961
70	Central Khmer	Khmer (F3)	8,404	672	15,919	28,557	44,449	36,360	24,338	52,252	98,010	53.31%	45,758
71	Lao	Kam-Tai (F8)	20,719	1,718	10,770	610	11,370	22,836	33,059	33,596	73,912	45.45%	40,316
72	Dari	Iranian (F11)	8,221	463	34,394	4,561	38,950	12,811	43,069	47,200	67,420	70.01%	20,220
73	Croatian	Slavic (F11)	4,525	6,859	804	971	1,746	12,258	12,173	13,033	79,634	16.37%	66,601
74	Assamese	Indic (F11)	257	707	531	37,514	38,044	38,228	1,274	38,758	48,917	79.23%	10,159
75	Tibetan	Bodic (F9)	35,994	10	4,717	1,300	6,017	36,496	40,717	41,213	42,449	97.09%	1,236
76	Romanian	Romance (F11)	1,263	3,822	2,082	1,075	3,143	6,127	7,140	8,195	82,190	9.97%	73,995
77	Sinhala	Indic (F11)	62	67	6,476	14,072	20,541	14,192	6,604	20,661	32,913	62.77%	12,252
78	Kirghiz	Turkic (F2)	31	2	8,331	110	8,421	141	8,362	8,452	31,536	26.80%	23,084
79	Scottish Gaelic	Celtic (F11)	158	4	1,512	0	1,512	162	1,674	1,674	16,686	10.03%	15,012
80	Dutch	Germanic (F11)	8	0	186	143	326	151	194	334	1,805	18.50%	1,471
81	Irish	Celtic (F11)	0	0	1,263	48	1,280	48	1,263	1,280	1,780	71.91%	500
82	Catalan	Romance (F11)	12	0	42	102	143	109	54	150	816	18.38%	666
83	Mongolian	Mongolic (F2)	4	10	429	63	490	76	443	503	647	77.74%	144
84	Swedish	Germanic (F11)	0	0	43	4	47	4	43	47	364	12.91%	317
85	Danish	Germanic (F11)	6	0	52	13	65	19	58	71	337	21.07%	266
						INTERNATIONAL							
86	Esperanto	Constructed (F11)	0	0	27	103	130	103	27	130	565	23.01%	435
						NORTH AMERICA							
87	Haitian	Creoles and Pidgins (F6)	5,890	12	8,346	3,240	11,582	9,118	14,246	17,460	26,009	67.13%	8,549
						PAPUNESIA							
88	Indonesian	Malayo-Sumbawan (F12)	57,358	7,899	131,349	81,850	213,191	146,982	196,586	278,323	492,909	56.47%	214,586
89	Filipino	Greater Central Philippine (F12)	5	0	40	52	92	57	45	97	294	32.99%	197
90	Tetum	Central Malayo-Polynesian (F12)	0	0	2	0	2	0	2	2	15	13.33%	13
91	Bislama	Creoles and Pidgins (F6)	3	0	0	0	0	3	3	3	4	75.00%	1
						SOUTH AMERICA							
92	Aymara	Aymaran (F0)	32	0	110	104	213	129	142	238	827	28.78%	589
		totals	2,981,925	1,775,581	10,315,099	5,145,760	15,238,148	9,404,789	14,891,856	19,497,177	31,940,180	58.04%	12,443,003

Paper IV

Abstractive Summarizers are Excellent Extractive Summarizers

Daniel Varab
Novo Nordisk
IT University of Copenhagen
djam@itu.dk

Yumo Xu
University of Edinburgh
yumo.xu@ed.ac.uk

Abstract

In this paper, we explore the efficacy of modeling extractive summarization with an abstractive summarization system. We propose three novel inference algorithms for sequence-to-sequence models, evaluate them on established summarization benchmarks, and show that recent advancements in abstractive designs have enabled them to compete directly with extractive systems with custom extractive architectures. We show for the first time that a single model can simultaneously produce both state-of-the-art abstractive and extractive summaries, introducing a unified paradigm for summarization systems. Our results question fundamental concepts of extractive systems and pave the way for a new paradigm - generative modeling for extractive summarization.[1]

1 Introduction

Extractive summarization selects a set of salient sentences from the original document(s) and composes them into a summary. Compared to abstractive summaries made up of words or phrases that do not exist in the input, extractive summaries are less creative but avoid inconsistencies and hallucinations. The pipeline for building an extractive summarizer typically consists of two separate stages: *sentence labeling* and *extractive modeling*. As most summarization datasets do not come with gold labels indicating which document sentences are summary-worthy, the first step is to extrapolate *oracle* sentence labels, e.g., with greedy search (Nallapati et al., 2017). The task can then be *modeled* with a sequence labeling architecture (Cheng and Lapata, 2016): a salience score is estimated for each document sentence, and top-ranked sentences are selected for summary inclusion. Recent work has also expanded extractive modeling to higher-order sentence selection to account for complex label dependencies, via extracting sentences stepwise (Narayan et al., 2020), or reranking a small set of summary candidates (Zhong et al., 2020).

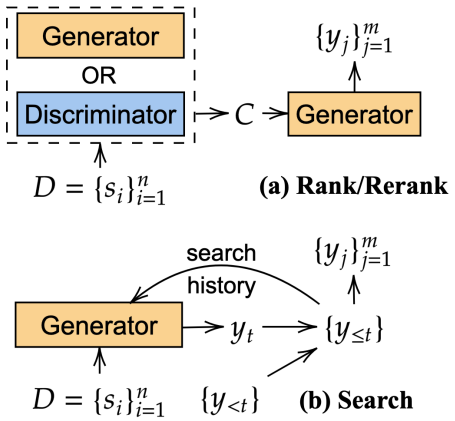


Figure 1: Proposed inference methods for GenX. We show (a) a two-stage approach with generative ranking/reranking, which creates a set of candidate summaries C from document sentences D, and (b) a single-stage inference method, generative search, which extracts summary sentences y_t autoregressively.

In this work, we revisit these fundamental concepts in extractive summarization. Specifically, we highlight that heuristically-derived sentence labels can be highly suboptimal (Narayan et al., 2018b; Xu and Lapata, 2022b), and customized neural architectures for extractive modeling prevent taking advantage of independent improvements. On the other hand, we recognize that generative modeling with a neural encoder-decoder architecture (Bahdanau et al., 2015; Sutskever et al., 2014), the *de facto* choice for abstractive summarization (Nallapati et al., 2017; Zhang et al., 2020; Lewis et al., 2020), constitutes to a promising solution for extractive summarization. Particularly, it learns directly from abstractive references and therefore does not require sentence labeling, and also embodies the extractive capabilities previously enabled by specialized neural architectures. Existing literature has established varied and many connections be-

[1]Our code will be available at https://github.com/anonymous/genx.

tween abstractive and extractive modeling such as copy mechanism (See et al., 2017), content selection (Kedzie et al., 2018; Gehrmann et al., 2018), and generation guidance (Dou et al., 2021). These connections, however, are mostly *abstract-centric* which are identified or constructed to improve abstractive summarization. In contrast, there are few studies from an *extract-centric* point of view.

In this work, we propose a new summarization paradigm that unifies extractive and abstractive summarization with generative modeling, *without* compromising abstractive performance. To this end, we treat extractive summarization as an *inference*-time task, and explore methods for adapting a pre-trained abstractive system for extractive summarization without further optimization. We assume that an abstractive system can be used as a summary evaluator for not only abstracts but extracts as well. Specifically, a model optimized only on abstractve references should be able to provide an accurate quality estimation for an extractive hypothesis, conditioned on the input document. A straightforward approach to validate this assumption is to search for the best document extract with an abstractive model for candidate evaluation. However, performing an exhaustive search over a combinatorial space of all eligible summary candidates is computationally intractable. To tackle this challenge, we propose GenX, **Gen**erative e**X**tractive summarization, which introduces a set of inference algorithms (shown in Figure 1) to reduce the search complexity via various approximations of the entire search space, at either sentence- or summary-level.

Experiments show that while retaining the ability to generate abstracts, GenX achieves competitive performance compared to systems developed only for extractive summarization on the CNN/DM benchmark. Particularly, in the one-stage summarization setting, it shows superior results than state-of-the-art systems. GenX also exhibits high robustness in zero-shot transfer: on XSum, its zero-shot performance surprisingly surpasses its fully supervised counterpart. We further conduct an extensive analysis of GenX's properties, providing potential directions for future research on generative modeling for extractive summarization.

2 Generative Modeling for Extracts

Given a generative model θ trained on summarization data comprising documents and abstractive references, at inference time, for an input document D and a summary sequence Y, we estimate the length-normalized log probability of Y, following the standard practice in neural text generation (Cho et al., 2014):

$$p_\theta(Y|D) = \frac{1}{|Y|^\alpha} \sum_{t=1}^{|Y|} \log p_\theta(Y_t|D, Y_{<t}) \quad (1)$$

where α is a length penalty term. As θ is optimized at token-level, we evaluate both *complete* and *partial* summaries with $p_\theta(Y|D)$.

The candidate summary space for a document $D = \{s_i\}_{i=1}^n$ of n sentences is combinatorial, consisting of $|\mathbb{C}(D)| = C\binom{m}{n}$ candidate summaries of length m. To sidestep the computational intractability, we introduce three inference algorithms that reduce the search complexity via approximations. The first two construct a candidate summary set, using either a discriminative or generative model (see Figure 1(a)), while the last approach directly searches over the partial summary candidate space (see Figure 1(b)).

Generative Ranking We infer a pre-trained generative model at both sentence- and summary-level for hierarchical ranking. Specifically, we input each document sentence s into a generator and evaluate its summary-worthiness independently via its likelihood. We then rank all document sentences, and any subset of size m of the top-k sentences is considered as a candidate summary c. A sequence-to-sequence generator then evaluates and ranks all candidate summaries, and the highest-ranked one is selected as the extractive hypothesis:

$$y = \underset{c \subseteq \text{top-k}\, p_\theta(s|D)}{\text{argmax}} \; p_\theta\left(\oplus(c)|D\right) \quad (2)$$

where $\oplus$ concatenates the selected document sentences in c, ordered by their rank.

Generative Reranking Instead of using the same generative model for both sentence and summary evaluation, we assume access to an existing discriminative model $p_\phi(s|D)$ for sentence evaluation and ranking. Following Zhong et al. (2020), we adopt BERTSUM (Liu and Lapata, 2019) to score each document sentence and then build candidate summaries as the combinations of top-scoring sentences. In this case, the role of generative modeling is a summary-level reranker $p_\theta(\oplus(c)|D)$.

Model	R-1	R-2	R-L
Lead-3	40.42	17.62	36.67
Oracle	52.59	31.23	48.87
One-Stage Systems			
BertSumExt	*42.73*	*20.13*	*39.20*
Stepwise ETCSum*	***43.84***	***20.80***	*39.77*
GenX (Search)	43.57	20.55	**40.01**
Two-Stage Systems			
BertSumExt+TRB	*43.18*	*20.16*	*39.56*
MatchSum	***44.41***	**20.86**	***40.55***
Posthoc Rank	*39.77*	*18.51*	*36.00*
GenX (Rank)	*42.90*	*19.99*	*39.09*
GenX (Rerank)	43.76	20.82	40.02

Table 1: Results on CNN/DM test set. We highlight **highest** scores, and scores of one-stage and two-stage systems that are *outside the 95% confidence interval* of GenX (Search) and GenX (Rerank), respectively (with 95% confidence interval via bootstrap resampling (Davison and Hinkley, 1997)).

Generative Search Instead of ranking, we consider how to construct a summary by directly searching over the *sentence* space, i.e., without first crafting several candidate summaries from the input document. We propose a novel search algorithm that autoregressively selects a sentence until a stopping criterion is satisfied. Specifically, at each search step t, we evaluate and select a sentence as:

$$y_t = \underset{s \in D}{\operatorname{argmax}}\, p_\theta(y_{<t} \oplus s | D) \quad (3)$$

where $\oplus$ concatenates the selected sentences $y_{<t}$ and a candidate sentence s. The selected sentence y_t is then concatenated with $y_{<t}$ to form the selection history for the next step, as shown in Figure 1(c). Narayan et al. (2020) also introduces a stepwise model which employs a special end token and the search stops when the token is generated. As an alternative, we follow the common practice in non-autoregressive extractive summarization (Liu and Lapata, 2019; Zhong et al., 2020) and assume a fixed number of sentences in the summary hypothesis, leading to a fixed number of search steps. We additionally experiment with a dynamic stopping criterion where the search over sentences continued until the end of sequence token `EOS` provides a higher summary likelihood than adding an additional sentence:

$$\text{s.t.} \max_{s \in D} p_\theta(y_{<t} \oplus s | D) > p_\theta(y_{<t} \oplus \texttt{EOS} | D). \quad (4)$$

Model	R-1	R-2	R-L
BertSumExt (ZS)	20.54	2.93	15.55
BertSumExt+TRB (ZS)	20.62	2.95	15.62
MatchSum (ZS)	20.90	**3.07**	15.75
GenX (Search; Supervised)	17.90	2.79	13.36
GenX (Search; ZS)	**20.94**	2.96	**15.92**

Table 2: Results on XSum test set. We highlight **highest** scores. ZS denotes zero-shot performance for models trained on CNN/DM while Supervised uses XSum for training.

3 Experimental Setup

We perform supervised experiments on CNN/DM (Hermann et al., 2015) and zero-shot experiments on XSum (Narayan et al., 2018a). We evaluate summaries with ROUGE (Lin and Hovy, 2003). Details for our experimental settings and datasets can be found in Appendix A.

As there are no established baseline for extractive summarization with generative modeling, we further construct **Posthoc Rank**, a posthoc method for direct comparison with GenX. Specifically, an abstract is firstly produced from a pretrained generative model. Then, the generated abstract is used to query a set of document sentences, and m sentences are retrieved with BM25 as the summary.

4 Results

Supervised Summarization Table 1 shows the results of various systems trained and evaluated on CNN/DM. The first block presents the performance of the Lead-3 baseline which considers the first 3 sentences in a document as the summary and Oracle which serves as an upper bound.

The second block reports the performance of one-stage summarization systems. Stepwise ETCSum (Narayan et al., 2020) is a state-of-the-art autoregressive system that learns to score partial summaries by selecting which sentence is a summary sentence iteratively. Different from GenX, it trains a highly-customized extractive architecture with extractive oracle summaries. As can be seen, GenX performs on par with Stepwise ETCSum, and significantly outperforms BertSumExt (Liu and Lapata, 2019), a popular extractive system based on sequence labeling, without requiring any extractive training.

The third block presents the results of two-stage systems. BertSumExt+TRB (Liu and Lapata, 2019) adds an additional stage for sentence selection with

Model	R-1	R-2	R-L
GenX (Search)	43.57	20.54	40.01
BART	↓5.11	↓4.12	↓5.08
Dynamic Stopping	↓0.11	↓0.08	↓0.10
Trigram Blocking	↓0.16	↓0.26	↓0.18
Ext. Candidates	↓2.57	↓1.59	↓2.54

Table 3: Ablation study in CNN/DM test set.

Trigram Blocking, an effective method for redundancy removal. MatchSum (Zhong et al., 2020) is a state-of-the-art extractive system that takes top-ranked sentences from BertSumExt and then reranks the summary candidates composed by them with a model based on a Siamese-Bert architecture. As can be seen, GenX models consistently improve over the single-stage BertSumExt, i.e., with or without BertSumExt as a sentence-level ranker. Its reranking variant also outperforms BertSumExt+TRB, showing that generative summary-level evaluation is more effective than heuristically-derived selection criteria. Note that the performance of GenX still falls short of state-of-the-art MatchSum. However, we note that GenX is built to retain the base model's ability in abstractive summarization, which is not applicable to any compared extractive systems (except Posthoc Rank, which shows significantly inferior performance).

Zero-Shot Summarization We further examine the generalization capability of extractive systems in a *zero-shot* setting.[2] As shown in Table 2, GenX generalizes to a different dataset robustly, outperforming strong one- and two-stage systems. It is generally perceived that a model's zero-shot performance is inferior to the supervised performance. Surprisingly, GenX performs substantially better in the zero-shot setting than its supervised counterpart. One potential reason is that despite the discrepancy between training and inference, CNN/DM is a more extractive dataset than XSum (Liu and Lapata, 2019), and therefore contains more extract-specific knowledge. Compared to existing systems, GenX is more capable of transferring the extractive ability learned from CNN/DM to XSum. This shows that treating extractive summarization as an *inference* task can significantly reduce the risk of overfitting to one specific dataset, shedding light on a new direction for knowledge transferring in zero-shot summarization.

[2]We did not include zero-shot results of Stepwise ETCSum as there are no publicly available code or models.

5 Ablation Study

We further assessed GenX with an ablation study. Replacing BRIO (trained with MLE and Contrastive Loss) with Bart (trained with MLE) leads to the largest performance drop. With the augmentation of contrastive learning, the abstractive system is competent in a dual role of both a generation and evaluation model, showing the importance of calibrating a generative model on its summary-level probability, even for its extractive inference.

The dynamic stopping mechanism introduced in Equation (4) performs on par with fixed-step search, showing that learning directly from abstracts is a promising way to teach models *when to stop* for summary extraction. GenX is also shown to be able to search for extractive summaries of less redundancy: its performance can *not* be further improved via incorporating Trigram Blocking. At last, we introduced extractive summaries into the training course of the abstractive model that GenX is built on, which also leads to performance degradation (see details in Section 8).

6 Related Work

There is a plethora of work on controlling different aspects of summarization, from content (Xu and Lapata, 2022a; Ahuja et al., 2022) to formats (Zhong et al., 2022). In this work, we offer efficient and effective control over the summary type (extract vs abstract). Recent work also investigates how to treat discriminative tasks such as information extraction and retrieval with generative modeling, and have its effectiveness for entities (De Cao et al., 2020) and string identifiers (Bevilacqua et al., 2022). Despite the resemblances with such tasks, extractive summarization with generative modeling remains under-explored.

7 Conclusion

In this paper, we explored the possibility of modeling extractive summarization with an abstractive system. We proposed three novel inference algorithms which repurposed a generative model for the extractive task. Our results showed that not only is extractive summarization feasible, but it can also directly compete with contemporary extractive systems. This work shows that extractive and abstractive paradigms can be unified through a sequence-to-sequence design, removing the need for oracle summaries and custom extractive model architectures.

8 Limitations

One potential way to improve the extractive performance of a generative system is to explicitly model the likelihood of *extracts* during training. Driven by this intuition, we investigate creating a mixture of extractive and abstractive candidates for contrastive learning in BRIO. Specifically, we obtain extractive candidates with beam labeling proposed in Xu and Lapata (2022b), while the abstractive ones are from the original BRIO training data. Nevertheless, as we can see, this mixing method hurts both BRIO's extractive and abstractive performance. However, it is noteworthy that extractive summary is important in a wider context, as shown in Section 4: reference summaries in CNN/DM are highly extractive and optimizing a model on these summaries therefore may have provided it with the task instruction needed for extractive summarization, albeit implicitly. We leave the study of a more effective extract-aware learning strategy for future study.

Furthermore, we emphasize that the conclusions drawn in this paper are based on results produced on English datasets from the news domain. Even though these datasets are established benchmark datasets for summarization it is imaginable that other domains and languages may have produced different evidence. Despite this, the results remain insightful as the results show that extractive summarization is in fact feasible with modern abstractive systems. In future research, we look forward to shedding light on the possibilities and limitations of the proposed methods in a broader context.

References

Ojas Ahuja, Jiacheng Xu, Akshay Gupta, Kevin Horecka, and Greg Durrett. 2022. ASPECTNEWS: Aspect-oriented summarization of news documents. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6494–6506, Dublin, Ireland. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, CA, USA.

Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Scott Yih, Sebastian Riedel, and Fabio Petroni. 2022. Autoregressive search engines: Generating substrings as document identifiers. In *Advances in Neural Information Processing Systems*.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 484–494, Berlin, Germany.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

Anthony Christopher Davison and David Victor Hinkley. 1997. *Bootstrap methods and their application*. 1. Cambridge university press.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2020. Autoregressive entity retrieval. In *International Conference on Learning Representations*.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. GSum: A general framework for guided neural abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4830–4842, Online. Association for Computational Linguistics.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, Brussels, Belgium. Association for Computational Linguistics.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, pages 1693—-1701, Cambridge, MA, USA.

Chris Kedzie, Kathleen McKeown, and Hal Daumé III. 2018. Content selection in deep learning models of summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1818–1828, Brussels, Belgium.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. 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. Association for Computational Linguistics.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Human Language Technology Conference of the North American*

Chapter of the Association for Computational Linguistics, pages 71–78, Edmonton, Canada.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. 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 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022. BRIO: Bringing order to abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2890–2903, Dublin, Ireland. Association for Computational Linguistics.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pages 3075–3081, San Francisco, California, USA.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018a. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018b. Ranking sentences for extractive summarization with reinforcement learning. 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 1747–1759, New Orleans, Louisiana.

Shashi Narayan, Joshua Maynez, Jakub Adamek, Daniele Pighin, Blaz Bratanic, and Ryan McDonald. 2020. Stepwise extractive summarization and planning with structured transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4143–4159, Online. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Yumo Xu and Mirella Lapata. 2022a. Document summarization with latent queries. *Transactions of the Association for Computational Linguistics*, 10:623–638.

Yumo Xu and Mirella Lapata. 2022b. Text summarization with oracle expectation. *arXiv preprint arXiv:2209.12714*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online. Association for Computational Linguistics.

Ming Zhong, Yang Liu, Suyu Ge, Yuning Mao, Yizhu Jiao, Xingxing Zhang, Yichong Xu, Chenguang Zhu, Michael Zeng, and Jiawei Han. 2022. Unsupervised summarization with customized granularities. *arXiv preprint arXiv:2201.12502*.

A Implementation Details

We show detailed data statistics in Table 4. For our GenX experiments, we use the BRIO system (Liu et al., 2022) as our underlying abstractive model. To replicate the BRIO system we run the published code repository associated with the paper. Specifically, we initialize a BART model with the Huggingface Models Hub checkpoint `facebook/bart-large-cnn` and fine-tune it with the provided configuration using the training scheme presented in the paper on both the CNN/Dailymail, and XSum dataset using the data distributed in said repository. We train the model with full precision on a single machine with four Tesla V100 GPUs for 30 hours and choose the checkpoint with the lowest cross-entropy (generative) loss term on a held-out validation set. Interestingly choosing the checkpoint with the lowest contrastive term produces poor results. Also, using mixed precision training doesn't appear to work.

To run the inference algorithms we initialize a BART system with different weights, either obtained through the above training procedure (BRIO) or the baseline `facebook/bart-large-cnn` checkpoint. We set the length penalty term $\alpha = 1$ which simplifies the expression to the arithmetic mean since we are not interested in penalizing summary length. For this computation we run the model

Datasets	CNN/DM	XSum
Language	En	En
Domain	Newswire	Newswire
#Train	287,084	203,02
#Validation	13,367	11,273
#Test	11,489	11,332
#Sentences in Extract	3	2

Table 4: Data statistics for extractive summarization.

under fp16 mixed precision to save memory, however, casting the model entirely to half-precision for inference does not appear to work.

We used standard parameter settings for all experiments: ROUGE-1.5.5.pl -c 95 -m -r 1000 -n 2 -a.

B License Information

The datasets used in this work, CNN/DM (Hermann et al., 2015) and XSum (Narayan et al., 2018a), are both released under the MIT License.

C System Output

Document: We spend a third of our lives asleep, but most of us don't pay attention to what our mind and body actually need during these resting hours in order to feel refreshed every day. The Sleep Health Foundation have released a study reporting that 30 percent of Australians complain about their lack of sleep on a daily basis. According to Chair Professor David Hillman, those misplaced hours of sleep must be paid back in order to be functional for the entire week. A study has outlined that 30 percent of Australians complain about their lack of sleep on a daily basis. The average adult needs around eight hours of sleep per night with a range of seven to nine. The average amount of sleep for an adult is around eight hours, with a range of seven to nine, the ABC have reported. Any less than six hours or any more than 10 hours is unusual for the standard person. Professor Hillman added that our sleep pattern is influenced by how much we are willing to compromise from the work week. 'A lot of us pay back a bit of that debt on the weekend but I think it's possible to exist in a sort of tolerable, sleep-restricted state,' he said. 'In other words you're not optimal, but you're still functional.' Pushing these sleep-debt boundaries can lead to micro sleeps in certain people. Therefore, the hours must be paid back to avoid an error rate in alertness tasks. Any less than six or any more than ten hours is unusual for the standard person. If power napping, it is important to get no more than 20 minutes or inertia will set in. In relation to a sleep schedule, Professor Hillman said the eight hours per night does not necessarily need to be consecutive. 'Interestingly enough, your slow wave sleep, is in the first four hours,' he said. 'Most adults, the most convenient way our particular society is organised is to have your eight hours in a continuous block overnight but that's not a necessary thing.' If choosing to break up your eight hours of sleep, napping throughout the day is the answer. Professor Hillman advises 20 minute power naps to avoid falling into deep sleep and suffering from inertia which makes you feel temporarily worse off. 'The longer naps, you get the sleep inertia but ultimately once you've got up, they sustain you better,' he said. Professor Hillman has also advised that if you are waking up tired and fatigued it could be due to sleep apnoea which is often associated with snoring.

Reference Summary: The Sleep Foundation study has shown that adults need 8 hours of sleep. According to the study, 30 percent of Australians say they lack sleep daily. Professor David Hillman said it's important to pay back our sleep debts. He also says sleep can be broken up as long as you get the first 4 hours. Power naps should not be longer than 20 minutes or inertia will set in.

BertSumExt: The Sleep Health Foundation have released a study reporting that 30 percent of Australians complain about their lack of sleep on a daily basis. The average adult needs around eight hours of sleep per night with a range of seven to nine. Any less than six hours or any more than 10 hours is unusual for the standard person.

MatchSum: The Sleep Health Foundation have released a study reporting that 30 percent of Australians complain about their lack of sleep on a daily basis. The average adult needs around eight hours of sleep per night with a range of seven to nine.

GenX (Search): A study has outlined that 30 percent of Australians complain about their lack of sleep on a daily basis. The average adult needs around eight hours of sleep per night with a range of seven to nine. According to Chair Professor David Hillman, those misplaced hours of sleep must be paid back in order to be functional for the entire week.

Table 5: Examples of system output on the CNN/DM test set. BertSumExt adds an unnecessary sentence highlighted in red. MatchSum removes this sentence, while GenX adds an additional sentence, highlighted in blue, which is reflected in the reference summary but missing from the other two system outputs.

Paper V

With Good MT There is No Need For End-to-End: A Case for Translate-then-Summarize Cross-lingual Summarization

Daniel Varab
Novo Nordisk
IT University of Copenhagen
djam@itu.dk

Christian Hardmeier
IT University of Copenhagen
chrha@itu.dk

Abstract

Recent work has suggested that end-to-end system designs for cross-lingual summarization are competitive solutions that perform on par or even better than traditional pipelined designs. A closer look at the evidence reveals that this intuition is based on the results of only a handful of languages or using underpowered pipeline baselines. In this work, we compare these two paradigms for cross-lingual summarization on 39 source languages into English and show that a simple *translate-then-summarize* pipeline design consistently outperforms even an end-to-end system with access to enormous amounts of parallel data. For languages where our pipeline model does not perform well, we show that system performance is highly correlated with publicly distributed BLEU scores, allowing practitioners to establish the feasibility of a language pair a priori. Contrary to recent publication trends, our result suggests that the combination of individual progress of monolingual summarization and translation tasks offers better performance than an end-to-end system, suggesting that end-to-end designs should be considered with care.

1 Introduction

Cross-lingual summarization (CLS) is the task of producing a summary of a text document that differs from the language it was written in, e.g. summarizing Turkish news or Danish product reviews in Hindi or English. This not only allows users fast access to information but also grants individuals access to information that is otherwise inaccessible. CLS is a challenging task as it must solve the challenges of both machine translation (MT) and summarization. There have historically been two approaches to the task;

- Pipeline designs (translate, summarize)
- End-to-end designs (sequence-to-sequence)

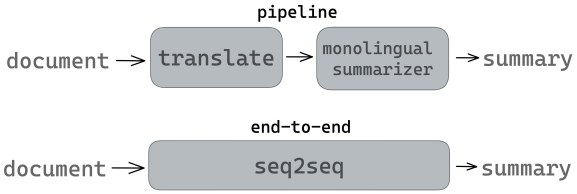


Figure 1: Pipeline versus end-to-end cross-lingual summarization designs. Pipeline-based systems perform cross-lingual summarization over two steps, first translating and then summarizing (or vice versa). End-to-end systems conflate translation and summarization by training a sequence-to-sequence to perform both tasks simultaneously.

Pipeline-based systems decompose CLS into two explicit steps, *translation* and *summarization*. This removes the necessity for parallel training data and enables taking advantage of ongoing innovations in translation and monolingual summarization research. The downside is the inherent effects of error propagation, where fx. a poor translation is forwarded to the subsequent summarization system, ultimately producing a bad summary. To circumvent this sequence-to-sequence designs have been proposed to avoid explicit translation and summarization steps altogether. With access to sufficiently large amounts of cross-lingual data, an end-to-end model can be trained to directly map an input document in one language, to a summary in another. The downside, however, is the sizable lack of CLS data, which does not occur naturally as opposed to the data of the implicit tasks: machine translation (Bañón et al., 2020; Aulamo and Tiedemann, 2019; Fan et al., 2021) and monolingual summarization (Hermann et al., 2015; Narayan et al., 2018; Grusky et al., 2018; Varab and Schluter, 2021; Hasan et al., 2021; Scialom et al., 2020). In spite of this, a growing body of research is pushing the envelope on end-to-end CLS systems. (Zhu et al., 2019) and (Cao et al., 2020) created large synthetic CLS datasets using back-translation for English and Chi-

nese. (Duan et al., 2019) proposed directly distilling a system from existing monolingual summarization and translation systems using teacher forcing. The latest efforts have been put into collecting CLS data from online websites (Ladhak et al., 2020; Perez-Beltrachini and Lapata, 2021; Bhattacharjee et al., 2021).

Contributions This paper investigates the immediate behaviors of two CLS paradigms on a wide range of languages and contributes with the following insights:

- End-to-end systems do not convincingly outperform simple pipeline systems (translate-then-summarize) - even if provided with large amounts of data.
- Provided with a competitive MT system, pipeline systems outperform strong end-to-end systems by a large margin.
- Publicly distributed BLEU scores are reasonably correlated with pipeline performance and can be used to estimate the efficacy of a language pair for CLS a priori.

2 Experiment

We wish to evaluate a paradigm's ability to perform CLS and to produce evidence that helps resolves the status quo. Let $D_s = [w_1, \ldots, w_n]$ be a text document consisting of words written in a source language s. The goal of a considered system is to produce a candidate summary S_t written in a target language t, such that S_t adequately summarizes the central information conveyed in D_s. In our experiments, we explore 39 different languages for s but fixate t = *English*. We run two recently proposed designs for end-to-end (E2E) CLS and compare them to two simple but performant pipeline systems. We choose *translate-then-summarize* (TTS) over *summarize-then-translate* (STT) because STT requires monolingual summarization systems for each language, while translation systems are available for most language pairs. Using TTS, therefore, allows us to investigate more languages while taking advantage of progress in monolingual summarization research, which is primarily developed for English. We also argue that English is a suitable target language as it aligns well well with the practical goals of cross-lingual summarization: knowledge sharing through trade and international languages (Guérard, 1922).

3 Models

3.1 Pipeline Systems

Having chosen TTS it is sufficient to find a single summarization system. Since the summarization system will be compared against a sequence-to-sequence model we choose an abstractive summarization which also builds on a sequence-to-sequence architecture. We choose the BRIO Liu et al. (2022) system as it has recently shown strong performance across several standardized summarization benchmark datasets. For translation, we consider two systems. First, we consider the OPUS-MT models (Tiedemann and Thottingal, 2020; Junczys-Dowmunt et al., 2018). OPUS-MT models are trained on the OPUS corpus (Aulamo and Tiedemann, 2019) and support 180+ languages. Secondly, to explore the difference if using a more powerful MT system we consider the 418M parameter M2M100 (Fan et al., 2021) model. This is a performant multilingual MT system that supports translation in any direction for 100 languages. We name these considered pipeline systems as follows:

TTS-weak combines the OPUS-MT translation system with the abstractive summarization system BRIO. This system intends to investigate the effects of a lightweight MT system and quantify the effects of poor translations, and the performance drops resulting from cascading errors.

TTS-strong combines the M2M100 translation system with the abstractive summarization system BRIO. This system acts as the competing alternative to an E2E system design. Results based on this system are the ones that will be considered when comparing the pipeline performance with E2E performance.

3.2 End-to-End

For end-to-end systems, consider the model proposed along with the CrossSum dataset (Bhattacharjee et al., 2021). This model proposes fine-tuning over multiple language simultaneously using a multistage sampling technique to account for imbalance across languages. They report that training on multiple languages improves the performance of the system as a result of knowledge sharing between related languages. We also consider a zero-shot cross-lingual model recently proposed by Perez-Beltrachini and Lapata (2021). This model is trained using monolingual English data but freezes

the embeddings and relies on the model to knowledge transfer to unseen languages. We adopt the described training scheme but refrain from incorporating the meta-learning loss as the authors only reported minor improvements compared to not using it. We name the considered E2E systems:

E2E-ZS is the latter zero-shot model proposed by Perez-Beltrachini and Lapata (2021). As text generation models are not known to transfer well to zero-shot settings, this system acts as a means to identify languages that are easy to transfer.

E2E-FT is the former fine-tuned model proposed by Bhattacharjee et al. (2021). This is a strong model with access to large amounts of data in multiple languages during training and, therefore, acts as an E2E system for CLS.

4 Dataset

We evaluate all systems on 39 languages in the validation set of CrossSum (Bhattacharjee et al., 2021), a large-scale cross-lingual summarization dataset containing news articles from the multilingual British news outlet BBC. CrossSum consists of 1.7 million document-summary pairs and more than 1500+ language pairs. The corpus is built on top of XL–Sum (Hasan et al., 2021), a multilingual extension to XSum (Narayan et al., 2018), and is created by aligning articles written in different languages using the multilingual sentence embeddings (Feng et al., 2022). CrossSumm contains summaries that like XL–Sum and XSum are short, often no longer than a single sentence.

5 Results

In Table 2 we report the results of our experiments. Each language is associated with an F-1 ROUGE-1 (Lin, 2004) and a BLEU score. We compute ROUGE scores with `sacrerouge` (Deutsch and Roth, 2020) using the default parameters[1]. The columns reflect the four considered models. The first three rows show average scores across subsets of languages filtered with BLEU scores. The rows provide detailed scores for each model on each language subset. ROUGE scores that are empty are due to the language not being supported, while empty BLEU scores are simply unavailable. We do include results whenever possible for completeness.

[1]ROUGE-1.5.5.pl -c 95 -m -r 1000 -n 2 -a

Language	**ROUGE-1**				**BLEU**
	TTS weak	TTS strong	E2E ZS	E2E FT	
Somali	-	23.3	18.3	**32.5**	97.6
Tamil	-	22.6	24.9	**30.7**	89.1
Ukrainian	38.1	**39.0**	25.7	33.5	64.1
Turkish	**42.2**	41.4	29.8	34.9	63.5
Russian	39.6	**40.1**	30.1	33.7	61.1
French	39.2	**39.3**	29.7	33.2	57.5
Sinhala	-	**33.4**	17.7	30.4	51.2
Arabic	38.2	**38.5**	23.1	32.4	49.4
Bengali	27.1	25.3	14.2	**29.4**	49.2
Marathi	13.6	**31.8**	16.0	29.1	47.8
Indonesian	**42.0**	41.8	28.9	35.5	47.7
Telugu	-	-	14.2	**29.4**	47.6
Thai	**32.7**	-	17.6	30.6	47.2
Portuguese	-	**36.8**	25.5	32.2	46.9
Spanish	34.9	**36.2**	27.8	31.4	46.4
Nepali	-	24.7	24.8	**32.2**	42.8
Japanese	34.8	**39.0**	30.1	35.3	41.7
Hindi	32.9	**39.5**	26.4	32.4	40.4
Korean	31.9	**34.4**	26.9	32.0	39.2
Igbo	22.4	26.7	15.9	**27.6**	38.5
Yoruba	17.5	20.4	18.2	**39.2**	36.3
Welsh	24.6	23.1	15.9	**31.6**	36.2
Hausa	18.9	23.7	17.3	**32.2**	35.7
Azerbaijani	21.4	28.5	20.0	**32.6**	30.4
Tigrinya	17.2	-	10.5	**20.3**	29.9
Panjabi	18.0	17.2	14.3	**27.7**	29.3
Oromo	11.9	-	10.7	**23.4**	27.3
Amharic	-	20.2	16.0	**30.1**	23.5
Persian	-	**37.5**	25.4	32.8	-
Scottish	-	15.5	16.7	**35.2**	-
Gujarati	-	11.9	13.9	**29.7**	-
Kirghiz	-	-	16.8	**34.8**	-
Burmese	-	14.2	20.4	**33.9**	-
Pushto	-	33.3	25.7	**33.7**	-
Rundi	29.0	-	19.4	**35.4**	-
Swahili	-	**38.3**	18.8	35.0	-
Urdu	18.0	21.6	17.1	**31.7**	-
Uzbek	-	17.0	17.9	**31.1**	-
Vietnamese	38.2	**42.0**	29.7	34.8	-

Table 1

Table 2: ROUGE-1 and BLEU scores for all four models, across all 39 languages. E2E_{ZS} denotes the E2E zero-shot system, E2E_{FT} the fine-tuned E2E system, TTS_{strong} the TTS system using the M2M100 translation system, and TTS_{weak}, the TTS system using the OPUS-MT translation systems.

Translation System Quality An obvious limitation of two-step systems is that poor translation systems are bound to produce poor-quality summaries. To quantify this relationship we search for

available BLEU test scores (Papineni et al., 2002) for translation-based systems for all investigated languages. We collect scores for the OPUS-MT systems, but could to our surprise only find scores on subsets of languages or aggregated scores over multiple languages for M2M100 and mBART50. For the lack of better, we report OPUS-MT BLEU test scores for each language and emphasize that conclusions based on these scores on other models should be taken with great care. We also acknowledge that BLEU is not comparable across datasets, however, we do argue that the scores may be used as an approximation for the quality of a translation system.

6 Analysis

The results reveal three central insights. First, it is clear from the results of E2E-ZS that zero-shot is not feasible for CLS on the CrossSum dataset. Second, E2E-FT produces mostly low-to-mid scores with little low variance across languages. This model has the highest mean of 31.9. Thirdly, TTS, despite having a slightly lower average of 28.5 and 29.6 between TTS-weak and TTS-strong respectively, these systems produce much higher scores on certain languages. A closer look reveals that despite E2E-FT scoring higher on average, both TTS systems frequently outperform E2Ef tune, and do so by a sizable margin. Conversely, when they do not they underperform significantly. Only four languages exhibit similar scores across the two paradigms, indicating a negative correlation between TTS-* and TT-FT. What we observe is that E2E-FT tune performs decently with little variation across languages, while TTS solutions either make or break it. Further inspection of the table suggests that the explanation for the TTS model's performance can be explained by low-quality translations. In Figure 2 we scatter plot translation and summarization scores for TTS systems and observe correlated behavior (Pearsons $\rho = 0.41$). A correlation that becomes visibly stronger if we allow removing suspicious BLEU scores (Somali and Tamil, $\rho = 0.75$).

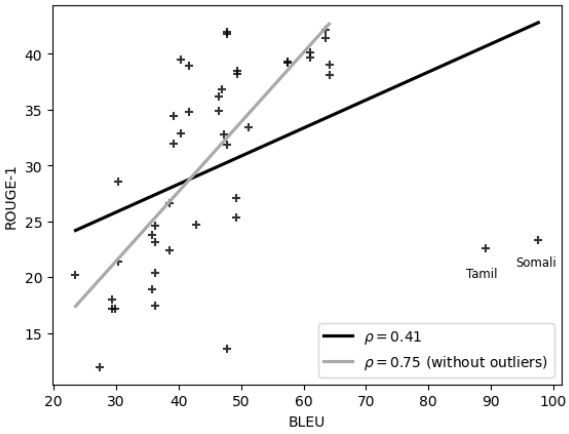


Figure 2: Collected BLEU scores on the x-axis and ROUGE-1 scores on the y-axis for TTS systems, including two outliers (Somali and Tamil) with suspiciously high BLEU scores. Removing the outliers further strengthens the relationship between the two metrics for TTS.

7 Conclusion

In this paper, we question the recent trends in favor of end-to-end system design for CLS and address the current lack of fair comparisons to pipeline-based methods. We evaluate these two paradigms on many-to-one CLS from 39 source languages into English and show that despite the recent claims, and a general push toward end-to-end models, pipeline-based models remain a strong candidate for the task. We analyze the performance of pipeline-based models and show that performance is strongly correlated with translation quality (according to BLEU), and emphasize that this can be used to aid the decision-making for the development of real-world systems a priori using only public resources. With the results presented in this paper, we have produced evidence that allows practitioners and future researchers to re-consider the benefits of pipeline-based models.

8 Limitations

The experiments presented in this paper revolve around a single dataset of a specific summary type (single-sentence summaries). It is possible to imagine that if the experiments were run on another dataset the results would have produced other conclusions. However, due to the scarcity of cross-lingual summarization data and no other sizable datasets, it is not unclear how to broaden the experiment while still having enough data to support training a sequence-to-sequence model. We believe the empirical evidence presented in this paper adds valuable insights to peers and practitioners in the NLP community and that these results may serve as a counterweight to the focus on end-to-end system designs, highlighting an increasingly overlooked model option.

References

Mikko Aulamo and Jörg Tiedemann. 2019. The OPUS resource repository: An open package for creating parallel corpora and machine translation services. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 389–394, Turku, Finland. Linköping University Electronic Press.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Uddin Ahmad, Yuan-Fang Li, Yong-Bin Kang, and Rifat Shahriyar. 2021. Crosssum: Beyond english-centric cross-lingual abstractive text summarization for 1500+ language pairs. *CoRR*, abs/2112.08804.

Yue Cao, Hui Liu, and Xiaojun Wan. 2020. Jointly learning to align and summarize for neural cross-lingual summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6220–6231, Online. Association for Computational Linguistics.

Daniel Deutsch and Dan Roth. 2020. SacreROUGE: An open-source library for using and developing summarization evaluation metrics. In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, pages 120–125, Online. Association for Computational Linguistics.

Xiangyu Duan, Mingming Yin, Min Zhang, Boxing Chen, and Weihua Luo. 2019. Zero-shot cross-lingual abstractive sentence summarization through teaching generation and attention. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3162–3172, Florence, Italy. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Çelebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2021. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22:107:1–107:48.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. 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 708–719, New Orleans, Louisiana. Association for Computational Linguistics.

Albert Léon Guérard. 1922. TF Unwin, Limited.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703, Online. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Faisal Ladhak, Esin Durmus, Claire Cardie, and Kathleen McKeown. 2020. WikiLingua: A new benchmark dataset for cross-lingual abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4034–4048, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022. BRIO: Bringing order to abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2890–2903, Dublin, Ireland. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Laura Perez-Beltrachini and Mirella Lapata. 2021. Models and datasets for cross-lingual summarisation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9408–9423, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506.

Nils Reimers. 2021. Easynmt-easy to use, state-of-the-art neural machine translation.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2020. MLSUM: The multilingual summarization corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8051–8067, Online. Association for Computational Linguistics.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Daniel Varab and Natalie Schluter. 2021. MassiveSumm: a very large-scale, very multilingual, news summarisation dataset. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10150–10161, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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. 2020. 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. Association for Computational Linguistics.

Junnan Zhu, Qian Wang, Yining Wang, Yu Zhou, Jiajun Zhang, Shaonan Wang, and Chengqing Zong. 2019. NCLS: Neural cross-lingual summarization. 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 3054–3064, Hong Kong, China. Association for Computational Linguistics.

A Appendix

A.1 Experimental Details

Abstractive Inference

All models considered in this paper involve one (E2E) or two generation steps (TTS) which involve a few choices and a set of hyperparameters. For translation we translate documents in their entirety, sentence-by-sentence using the library EasyNMT[2] (Reimers, 2021) which conveniently wraps the translation models considered in this work. We faced some issues with sentence segmentation in a few languages but changed the library code to make it work. For all summarization systems (including E2E) we truncate input documents to 512 tokens for all languages, use a beam size of 2, sample no longer than 128 tokens, and employ trigram blocking. When required by the model we add a decoder start token for English.

Training of Zero-Shot Model

To train the zero-shot model described in the model section we adopt the methodology proposed by Perez-Beltrachini and Lapata (2021) and implement it using Huggingface's `transformers` (Wolf et al., 2020), `DeepSpeed` (Rasley et al., 2020), and of course `PyTorch` (Paszke et al., 2019). We freeze the embeddings of the encoder and decoder of `mBART50` but do not prune the vocabulary. We also do not apply the proposed meta-learning algorithm LF-MALM for the sake of simplicity. We train the model with cross-entropy for 50.000 steps with a batch size of 32 using fp16 mixed-precision training and evaluate and save the model every 1000 steps. We also run a linear learning rate scheduler with warmup for 5000 steps (5e-5). Results are produced using the model with the lowest loss (1.886). This model took approximately 3 days to run on two NVIDIA T4 Tensor Core GPUs using `DeepSpeed`.

[2] `github.com/UKPLab/EasyNMT`