Abstractive Document Summarization in High and Low Resource Settings

DISS. ETH NO 26717

Abstractive Document Summarization in High and Low Resource Settings

A thesis submitted to attain the degree of
DOCTOR OF SCIENCES OF ETH ZURICH

presented by:

Nikola Ivanov Nikolov

MSc, ETH Zürich and University of Zürich
born on 02.09.1991
citizen of Bulgaria

May, 2020

Accepted on the recommendation of:

Prof. Dr. Richard H.R. Hahnloser, examiner

Prof. Dr. Martin Volk, co-examiner

Dr. Katja Filippova, co-examiner

Abstract

Automatic summarization aims to reduce an input document to a compressed version that captures only its salient parts. It is a topic with growing importance in today's age of information overflow.

There are two main types of automatic summarization. **Extractive** summarization only selects salient sentences from the input, while **abstractive** summarization generates a summary *without* explicitly re-using whole sentences, resulting in summaries are often more fluent.

State-of-the-art approaches to abstractive summarization are data-driven, relying on the availability of large collections of paired articles with summaries. The pairs are typically manually constructed, a task which is costly and time-consuming. Furthermore, when targeting a slightly different domain or summary format, a new parallel dataset is often required. This large reliance on parallel resources limits the potential impact of abstractive summarization systems in society.

In this thesis, we consider the problem of abstractive summarization from two different perspectives: high-resource and low-resource summarization.

In the first part, we compare different methods for data-driven summarization, focusing specifically on the problem of generating long, abstractive summaries, such as an abstract for a scientific journal article. We discuss the difficulties that come with abstractive generation of long summaries and propose methods for alleviating them.

In the second part of this thesis, we develop low-resource methods for abstractive text rewriting, first focusing on individual sentences and then on whole summaries. Our methods do not rely on parallel data, but instead utilize raw non-parallel text collections.

In overall, this work makes a step towards data-driven abstractive summarization for the generation of long summaries, without having to rely on vast amounts of parallel, manually curated data.

Zusammenfassung

Die automatische Zusammenfassung zielt darauf ab, ein Eingabedokument auf eine komprimierte Version zu reduzieren, die nur seine wesentlichen Teile erfasst. Es ist ein Thema mit wachsender Bedeutung im heutigen Zeitalter der Informationsüberflutung.

Es gibt zwei Haupttypen der automatischen Zusammenfassung. Bei der extraktiven Zusammenfassung werden nur die hervorstechenden Sätze aus der Eingabe ausgewählt, während bei der abstrakten Zusammenfassung eine Zusammenfassung ohne explizite Wiederverwendung ganzer Sätze erstellt wird, wodurch die Zusammenfassungen oft flüssiger sind.

Moderne Ansätze zur abstrakten Zusammenfassung sind datengesteuert und stützen sich auf die Verfügbarkeit großer Sammlungen von Artikeln gepaart mit entsprechenden Zusammenfassungen. Die Paare werden in der Regel manuell erstellt, eine Aufgabe, die kostspielig und zeitaufwendig ist. Außerdem ist bei der Anpassung hin zu einem anderen Bereich oder zu einem anderen Zusammenfassungsformat oft ein neuer Datensatz erforderlich. Diese große Abhängigkeit von spezialisierten Datensätzen begrenzt die potenziellen Anwendungen von abstrakten Zusammenfassungssystemen in der Gesellschaft.

In dieser Arbeit betrachten wir das Problem der abstrakten Zusammenfassung aus zwei verschiedenen Perspektiven: ressourcenintensive und ressourcenarme Zusammenfassung.

Im ersten Teil vergleichen wir verschiedene Methoden der datengetriebenen Zusammenfassung, wobei wir uns speziell auf das Problem der Erzeugung langer, abstrakter Zusammenfassungen, wie z.B. einer Zusammenfassung für einen wissenschaftlichen Zeitschriftenartikel, konzentrieren. Wir erörtern die Schwierigkeiten, die mit der abstrakten Erstellung langer Zusammenfassungen einhergehen, und schlagen Methoden zu deren Milderung vor.

Im zweiten Teil dieser Arbeit entwickeln wir ressourcenschonende Methoden für das abstrakte Neuschreiben von Texten, wobei wir uns zunächst auf einzelne Sätze und dann auf ganze Zusammenfassungen konzentrieren. Unsere Methoden stützen sich nicht auf gepaarte Datensätze mit vorhandenen Zusammenfassungen, sondern nutzen rohe Textsammlungen.

Insgesamt macht diese Arbeit einen Schritt in Richtung einer datengesteuerten abstrakten Zusammenfassung für die Erstellung langer Zusammenfassungen, ohne sich auf große Mengen gepaarter, manuell kuratierter Daten stützen zu müssen.

Acknowledgements

I thank my thesis advisor, Prof. Dr. Richard Hahnloser, for giving me the opportunity to pursue a Ph.D. in a field that I am very passionate about, and for his support and guidance. I am grateful for the flexibility I had, to work on a wide range of exciting topics and projects, to define and execute on my work, to learn, and to grow as a researcher. I also thank my co-examiners, Dr. Katja Filippova and Prof. Dr. Martin Volk, for agreeing to be on my thesis committee, and for their helpful feedback on my work.

During my Ph.D., I had the pleasure of supervising eleven student projects. I enjoyed working with a diverse range of people from different backgrounds, guiding them, and learning a lot myself in the process. I thank the students for their dedication and hard work, especially the three students who did their Master theses with me: Yingqiang Gao, Alessandro Calmanovici, and Patrick Frank. I am also thankful for the many talented people I got the chance to collaborate with: Yuhuang Hu, Eric Malmi, Curtis Northcutt, Loreto Parisi, Jonathan Prada, Onur Gokce, Nianlong Gu, and others. I enjoyed working with all of you, and I wish you all the best in the future!

Finally, I thank everyone at the Institute of Neuroinformatics, for providing a friendly work environment and good vibes, and my parents for all of their support.

Nikola Nikolov, March 2020

Contents

Chapter 1

Introduction

Automatic summarization is a task in Natural Language Processing (NLP) concerned with the development of algorithms that can reduce an input text, such as a scientific journal article, to a compressed version that contains only the critical information, such as an abstract or a press release.

This thesis considers different approaches to text summarization of whole documents. The approaches fall into two broad categories: methods that rely on high amounts of parallel data for training (collections of articles with summaries manually written by humans), and techniques that are also data-driven but relax the requirement for having large amounts of parallel training data.

The thesis is organized as follows. This chapter begins by introducing relevant background concepts in NLP. Following that, we present the problem of abstractive text summarization and formalize it as a problem part of the family of structured prediction problems. In Chapter 2, we give an overview of the main contributions of this thesis. Chapters 3 and 4 contain the main content of the thesis, while Chapter 5 concludes the thesis and discusses possible directions for future research.

1.1 Natural Language Processing and Deep Learning

Natural Language Processing (NLP) is a field at the intersection of Linguistics, Computer Science, and Machine Learning, concerned with the development of computational techniques for the automatic analysis, understanding, and generation of human language content (Cambria and White, 2014). Some example applications are translating sentences from one language to another (Koehn, 2009; Castilho et al., 2017), dialogue systems for automatic conversing with users (Serban et al., 2016; Adiwardana et al., 2020), and speech-to-text conversion.

The vision of NLP is to build machines that can intelligently read, understand and integrate large quantities of human knowledge and help us with mundane tasks related to these (Hirschberg and Manning, 2015), such as giving us a personalized summary of a book. NLP is becoming especially important in our current age of big data, where the accumulation of textual information available on the web has been increasing tremendously in the last years, not only for general knowledge such as Wikipedia or news articles but also in the scientific literature (Hunter and Cohen, 2006).

Early research in NLP focused on the task of defining manual rules for the transformation of linguistic units. This task proved to be challenging, because of the high variability and ambiguity of human language, and led to the focus on *statistical* approaches, that have become prevalent in the last decades (Jurafsky and Martin, 2008). Many of these methods rely on sparse lexical features, such as co-occurrence statistics of windows of words (n-grams), part-of-speech sequences, and others. Such features can be fed to a standard shallow linear classification algorithm, such as a support vector machine or logistic regression. Crucially, success in NLP has been driven by the availability of large amounts of training data.

In recent years, **deep neural networks** (Goodfellow et al., 2016) have revolutionized many areas of machine learning, including NLP. They have achieved this through learning intermediate representations over the input data, through multiple layers of stacked units, which are optimized against a top-level loss function (LeCun et al., 2015). The parameters of a deep learning model learn to self-organize in an end-to-end fashion to minimize this loss. This approach reduces the need for task-specific feature engineering - little or no *a priori* linguistic domain knowledge is required to apply deep neural networks to a novel task (Collobert et al., 2011).

The success of deep models has lead to their application to a number of tasks in NLP, such as sentiment analysis (Socher et al., 2013), image caption generation (Xu et al., 2015a), Neural Machine Translation (NMT) (Bahdanau et al., 2015; Cho et al., 2015; Vaswani et al., 2017) and representation learning (Devlin et al., 2019). Furthermore, there are many potential applications of deep neural networks in bioinformatics and biomedical language processing (Mamoshina et al., 2016), e.g., for automated drug discovery or for navigating through the immense unstructured data available in the scientific literature.

1.2 Automatic Summarization

In this thesis, we deal with the problem of **automatic summarization**, which aims to reduce an input text and summarize it by a shorter, compressed version that captures the most relevant parts of the input, without any loss of essential information (Nenkova et al., 2011). For example, we may want to automatically generate a title for a scientific journal article, given its abstract. Automatic summarization is an area with a significant potential impact in today's age of information overflow, where over 2 million blog posts are written per day[1], and more than 1 million scientific articles are published every year (Björk et al., 2008). It is also an essential step towards building machines capable of understanding natural language.

Summarization is a difficult task even for people, and many professionals in the industry write summaries as part of their job (Mani, 2001). This difficulty comes from the fact that summarization requires multiple complex natural language understanding components that involve the selection, evaluation, aggregation, and reorganization of information, followed by its compression, generalization, and/or paraphrasing. Further to that, this needs to occur on multiple (possibly abstract) levels, such as sentence, paragraph, section, and document. Modern summarization systems are currently far from reaching human-like performance and fluency on this task. However, they might still provide useful summaries that can be time-saving (Nenkova et al., 2011). Evaluating the quality of output summaries (Section 1.2.4) is an important issue because of the subjective nature of this task, where multiple summaries can be deemed acceptable.

Summarization has resulted in a number of practical applications. Early examples include the summarization tool part of Microsoft Word (Yih et al., 2010), systems that summarize voicemail messages for users, to help them determine the priority of a call (Koumpis and Renals, 2005), as well as systems that can provide a digest of user forums (Nenkova and Bagga, 2004). More recently, summarization systems have been applied to generate headlines of news articles automatically (Rush et al., 2015) or to generate summaries of threads on Reddit[2].

1.2.1 Types of automatic summarization

A wide range of summarization variants can be differentiated, depending on the desired types of input and summary (Nenkova et al., 2011). First, summarization systems differ in their input and output **format**. Summarization of single documents is most frequent; however, multi-document summarization (Lin and Hovy, 2002; Lebanoff et al., 2018), which provides a single digest for multiple documents, or a whole topic, is becoming more common. In this thesis, we focus on single-document summarization.

The output format of the generated summary can also differ in size (e.g., a full paragraph vs. a single sentence, a headline, or several keywords). The

[1] https://wordpress.com/activity/posting/

[2] https://www.reddit.com/user/autotldr

content of the summary produced can be either broad (provide an overview or ”aboutness” of a document) or specific, containing only the critical information. Further to that, extra constraints can be added to produce more specific summaries, e.g., a user query or information about the target audience.

Concerning the **form** of the summaries generated, two types can be identified: *extractive* or *abstractive* (Nenkova et al., 2011; Mani, 2001) summarization. **Extractive summarization** extracts and combines sentences and phrases exactly as they appear in the original document. Thus, the output is a compressed and reordered version of the input, with the original wording usually remaining the same. In **abstractive summarization**, the output is typically generated using a language model trained on a collection of summaries, which is further conditioned on the input text. Thus, summaries may contain fragments that are not part of the original. Abstractive summarization results in more natural, human-like summaries, which may reference concepts that are not explicitly mentioned in the original document. However, extracting specific information is an equally important component, and combining the two has been shown to give an advantage in certain cases (Rush et al., 2015; Chen and Bansal, 2018a).

A final distinction can be made between **generic** and **domain-specific** approaches to summarization. The former aims to develop a single approach for any type of input text, whereas the latter leverages information available in a particular domain to improve summarization quality. This information could be related to the specific style of writing or document structure, which has been previously used for scientific article summarization (Teufel and Moens, 2002).

Tasks related to Automatic Summarization

There are several monolingual text rewriting tasks which are closely related to automatic document summarization, and which could also be useful components within a summarization pipeline. We introduce three of those tasks below.

Sentence paraphrasing is the task of rewriting an input sentence into a new form that has the same meaning as the input (Ganitkevitch et al., 2013; Napoles et al., 2016), without any removal of content, or change of the language. Paraphrasing can be a useful tool for generating alternative expressions of a text, that can be used as an intermediate step in a text generation or a data augmentation pipeline, for example.

Sentence compression (Knight and Marcu, 2002; Rush et al., 2015; Filippova et al., 2015) is similar to paraphrasing, however it adds the additional constraint that the output paraphrase should be shorter, while still maintaining the same meaning as the input. Sentence compression is also similar to summarization, however it operates on the sentence rather than the document level.

Text simplification (Alva-Manchego et al., 2020) aims to modify an input sentence to make it easier to understand. Simplification can involve a number of operations, such as lexical substitution, sentence splitting, reordering of the information, or removing peripheral information. Although simplification is not a specific goal of automatic summarization, it could be useful for adopting generated summaries towards a specific target user group.

1.2.2 Approaches to summarization

Early research in **extractive** summarization focused on *unsupervised* sentence ranking (Nenkova et al., 2011). Systems *rank* and select several sentences from the input using linguistic features (e.g., the probability of occurrence of a word in the input) that determine the importance of each sentence. Selection is typically followed by several post-processing steps that "stitch together" the sentences, to produce the final summary.

More recently, supervised statistical approaches have begun to be applied to summarization, e.g., systems based on statistical machine translation (Wubben et al., 2012) or sentence compression[3] (Filippova et al., 2015). In the seminal work of (Rush et al., 2015), they apply an **end-to-end** learning approach to **abstractive** summarization, inspired by the recent success of deep sequence learning models for Neural Machine Translation (NMT). All of the components of their system are trained in a data-driven manner, using pairs of input and output summaries. Their system can generate an abstractive headline of a news article from its first sentence, using a minimal amount of linguistic information. They compare their system to several classical approaches, to achieve superior performance on the Gigaword and DUC datasets, which are frequently used benchmarks for summarization (see Section 1.2.3).

1.2.3 Datasets for summarization

Supervised statistical approaches to summarization rely on the availability of large and high-quality datasets of article-summary pairs. Collecting such datasets is both expensive and time-consuming since manual human annotation is usually necessary to produce high-quality example summaries.

[3]The problem that deals with finding a shorter paraphrase of a single sentence by deleting unimportant parts.

Dataset	Article	Summary
Gigaword	russian defense minister ivanov called sunday for the creation of a joint front for combatting global terrorism	russia calls for joint front against terrorism
CNN/DM	(truncated): they are supposed to be the dream team who can solve the conundrum of how to win the election . but david cameron and boris johnson were left scratching their heads today as they struggled with a children 's jigsaw teaching toddlers about the seasons . as the london mayor tried to hammer ill-fitting pieces together with his hands , the prime minister tried out what could be a new election slogan , telling him : ' if in doubt , wedge it in . ' after being put right by a four-year-old who spotted their errors , the pair had more fun finger painting with tory blue paint . david cameron and boris johnson were left stumped by the puzzle at advantage children 's day nursery in surbiton , as three-year-old stephanie looked on . when they tried to put the puzzle back together , they hit trouble after it proved to be more difficult than expected . the conservative duo made their first appearance together on the campaign trail with a visit to advantage day nursery in surbiton , south west london . they were supposed to be highlighting tory plans to double free childcare for 600,000 children aged three and four . they sat at a table with joshua and stephanie , both aged three , and four-year-olds hamish and leo where they broke up a puzzle which depicted the four seasons . when they tried to put it back together , they hit trouble after it proved to be more difficult than expected .	david cameron and boris johnson visit nursery to promote childcare . duo left flummoxed by jigsaw aimed at teaching children about seasons . london mayor resorted to brute force to hammer wrong pieces together . prime minister gave his approval , telling boris : ' if in doubt , wedge it in ' .

Table 1.1: Example articles and summaries from two parallel summarization datasets that are frequently used in the recent literature on summarization.

For single-sentence summarization, the tasks from the Document Understanding Conference (DUC)[4] in 2003 and 2004, which are based on 500 news articles, have been frequently used as a benchmark. More recently, researchers have begun to leverage the availability of the large amount of textual data freely available on the web (Hermann et al., 2015), or provided by the Linguistic Data Consortium[5](Rush et al., 2015), to generate novel large-scale datasets for NLP problems, including for automatic summarization. In (Rush et al., 2015), as well as in other subsequent research, e.g., (Nallapati et al., 2016; Chopra et al., 2016; Malmi et al., 2019), the Annotated Gigaword Corpus (Napoles et al., 2012) of news articles has been established as a standard benchmark for abstractive sentence compression. The first sentence of each article was paired with its headline, to produce a large dataset of over 4 million training pairs (Table 1.1 contains an example).

When it comes to summarization on the document level, there are currently few large-scale datasets available. The CNN/Dailymail (CNN/DM) dataset (Hermann et al., 2015) consisting of pairs of news articles and short summaries in the form of abstractive bullet points (4 sentences on average, manually written for each article) has been a popular choice in recent research (See et al., 2017a; Narayan et al., 2018a; Gehrmann et al., 2018; Chen and Bansal, 2018a). When it comes to generating longer summaries, such as paragraphs longer than five sentences, there are few large-scale resources available. This lack of parallel data poses a demand for creating novel such datasets, which we consider in Chapter 3.1. Furthermore, the majority of summarization efforts are in the news domain, motivating the creation of datasets for other domains, such as for scientific journal articles.

1.2.4 Summarization evaluation

Since the early days of automatic text summarization, evaluation has been an essential issue (Mani, 2001). The open-ended and subjective nature of the task makes the practical evaluation of summarization systems particularly challenging. Manual evaluation using human raters is currently the most rigorous approach to summarization evaluation because it directly measures the expectations of real-life users. However, manual evaluation is often very time-consuming and expensive to perform, provoking researchers to invent automatic metrics that aim to approximate human judgments.

Below we briefly discuss the most popular methods for human and automatic evaluation of summarization systems. Hardy et al. (2019) contains a more in-depth and comparative discussion of a wide range of summarization evaluation approaches.

[4] `http://duc.nist.gov/`
[5] `https://www.ldc.upenn.edu/`

Human evaluation

Manual evaluation of summaries typically involves defining a set of questions that are posed to a cohort of human raters. Some questions may measure qualitative aspects of the generated summaries, such as their fluency or grammaticality, while others may measure the amount of meaning preservation from the article to the summary. Often, an absolute numerical scale (Likert, 1932), e.g., between 1 and 5, is used for such analysis, although preference-based evaluations that directly ask subjects to choose between two summarization outputs are also frequent.

Automatic evaluation

A good automatic metric for summarization needs to rank the quality of the information content that was selected, as well as potentially the fluency of the output summary. Evaluating abstractive text creates additional challenges since the output summary may contain words that are not part of the input article. Despite these challenges, there are several automatic metrics that provide a quick way to objectively compare different summarization approaches, and give some insight into the quality of the generated summaries.

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) (Lin, 2004) is a recall-based metric which is commonly used for summarization evaluation because it achieves good correlation with human raters. There are several variants of ROUGE, with two variants frequently reported in the literature:

- **ROUGE-N** is based on the number of N-gram co-occurrences between a system summary and a gold standard summary, normalized by the sum of all n-grams occurring in the gold standard. ROUGE-1 and ROUGE-2 are frequently reported in the literature.

$$ROUGE_n = \frac{\sum_{gram_n \in input} count_{match}(gram_n)}{\sum_{gram_n \in input} count(gram_n)} \tag{1.1}$$

- **ROUGE-L** measures the length of the longest overlapping subsequence of words in each summary-gold pair (Lin, 2004).

An obvious limitation of ROUGE is that it computes a hard lexical overlap with the gold standard summaries. Therefore, abstractive systems might be penalized if they don't use the same vocabulary as the gold standard summary. One way to alleviate this limitation is to compute ROUGE over multiple gold standard summaries per article, given that those are available.

Bilingual Evaluation Understudy (BLEU) (Papineni et al., 2002) is a commonly used metric for automatic evaluation in Machine Translation (MT) systems. It is precision-oriented, counting the percentage of n-gram overlaps between candidate translations and gold standard, at the exact location in the two sequences. Although BLEU is not as commonly used for summarization evaluation, it could be useful as a supplement to ROUGE.

METEOR (Banerjee and Lavie, 2005) is another precision-based metric commonly used in MT evaluation. METEOR works by aligning system-produced sentences to a gold-standard sentence, calculating sentence-level similarity scores based on exact, stem, synonym or paraphrase matching of words and phrases. Studies of METEOR have found that it correlates better with human translators than BLEU. METEOR has been used in several recent automatic summarization studies to provide a different perspective from ROUGE.

SARI (Xu et al., 2016) is a recently proposed metric that additionally takes into account the total number of changes (additions, deletions) from the input text to the output. Although SARI was originally developed for sentence simplification, it can also provide insightful results for sentence as well as for document summarization.

1.2.5 Scientific article summarization

One domain of particular relevance to this thesis is scientific journal articles. The rate of scientific journal publications has been growing exponentially each year (Hunter and Cohen, 2006). Thus, keeping track of the latest research and identifying impactful work has been becoming increasingly difficult, especially for younger researchers, creating a demand for the development of automated computational tools and techniques for scientific textual mining and analysis. Efforts in this area include literature-based approaches for the discovery of novel drug compounds, retrieving articles containing specific information relevant to a user query, or automatic summarization of single or multiple articles within a field.

In previous research on scientific journal article summarization (Teufel and Moens, 2002; Nenkova et al., 2011, ch. 5) *generic* approaches (e.g., sentence ranking) performed much worse in comparison to domain-specific methods, manually engineered to exploit properties of the input. For example, scientific articles follow a specific layout structure that is usually common for all papers in a journal, and typically includes sections such as abstract, introduction, conclusion. Scientific texts are normally written in a canonical linguistic style, e.g., ”In this paper, we show that” indicates that important general information is about to follow. In (Teufel and Moens, 2002), they used such linguistic constraints to produce short scientific summaries. In other work (Qazvinian and Radev, 2008), the mentions and citations of a paper in other articles were used as a prime indicator of important content and were used to generate a summary of the impact of that paper in subsequent research. Cohan and Goharian (2018) utilize citations as well as the scientific discourse structure to perform extractive scientific article summarization.

An exciting frontier in scientific article summarization research is automated science journalism (Tatalovic, 2018). Recent studies have shown that the readability of scientific papers has been decreasing over time (Plavén-Sigray et al., 2017) due to an increase in general scientific jargon over the last decades. Automatic summarization technology could be useful for generating simplified summaries from papers, which could make science not only more accessible to non-experts but could also help to facilitate cross-disciplinary communication. In Chapter 4.2 of this thesis, we propose a new approach to synthesizing press releases from scientific journal articles.

1.2.6 Low-resource Summarization

Many recent methods for automatic summarization are domain- and language-independent, meaning that, given a suitable dataset in a new language such as German, they could potentially be successfully retrained on the new dataset out-of-the-box, or with only small hyperparameter modifications. However, the size and quality of the training dataset are crucial for achieving similar performance in the new target language. For many cases, such as for languages other than English, the availability of training examples can be scarce. This has motivated the development of alternative approaches to summarization that can still perform reasonably when applied in such low-resource conditions.

Early work on low-resource summarization predominantly focused on extractive approaches, which can be implemented using unsupervised heuristics (Nenkova et al., 2011). More recently, approaches that produce synthetic training datasets have been gaining popularity. For example, Parida and Motlicek (2019) leverage an auxiliary corpus in German for improving German text summarization (Parida and Motlicek, 2019). Another approach is to use the more abundant summarization resources in English to synthesize data in other languages, for example by using pre-trained machine translation systems, for which training data is generally much more readily available. In studies that take this approach, e.g., (Ouyang et al., 2019; Zhu et al., 2019; Nguyen and Daumé III, 2019), authors translate both articles and summaries from English summarization datasets, to construct summarization corpora for low-resource languages. This approach has been shown to be effective even for very resource-scarce languages, such as Somali, Swahili, or Tagalog.

An alternative approach to low-resource summarization that can still leverage large-scale datasets for pre-training is to cast summarization as a text-editing task (Malmi et al., 2019; Mallinson et al., 2020), rather than a full text rewriting task. Instead of generating a whole summary from scratch, this approach applies a set of edit operations to an input article, such as deleting, reordering, or inserting whole words or sentences.

1.3 Methods in Natural Language Processing

This section reviews important methods and concepts in NLP that are relevant for this thesis.

1.3.1 Text embeddings

One of the main reasons for the recent success of deep learning models in NLP is the introduction of dense representations, or **embeddings**. Text embeddings map units of text to a vector in high dimensional, but compact space of dimensionality d (e.g., of size 100-500). Word embedding methods (Mikolov et al., 2013; Pennington et al., 2014) focus on learning representations of individual words from large quantities of unstructured text, typically relying on a self-supervised training function, such as predicting a word given its surrounding words. Sentence embeddings (Le and Mikolov, 2014; Pagliardini et al., 2018; Devlin et al., 2019) focus on modeling whole sentences. Text embeddings are useful for a number of tasks such as question answering (Rajpurkar et al., 2016), natural language inference (Bowman et al., 2015), nearest neighbour retrieval (Kusner et al., 2015) and text alignment (Nikolov and Hahnloser, 2019).

1.3.2 Sequence modeling with neural networks

Many problems in NLP, including automatic summarization, deal with *sequences* of structures of *variable* length, such as words, subword units (Sennrich et al., 2016c), or even characters (Lee et al., 2017; Nikolov et al., 2018b). One such problem is **language modeling**, which lies at the core of many other problems that involve any form of language generation. Given a sequence $\boldsymbol{y} = \{y_1, ..., y_i, ..., y_t\}$ of t words, a language model estimates the probability distribution for the sequence $p(\boldsymbol{y}) = \prod_{i=1}^{t} p(y_i | \{y_1, ..., y_{i-1}\})$.

Recurrent Neural Networks (RNNs)

One approach to language modeling is to use a Recurrent Neural Network (RNN) to estimate this probability distribution (Mikolov et al., 2010). An RNN can perform recursive computation over each element of a sequence,

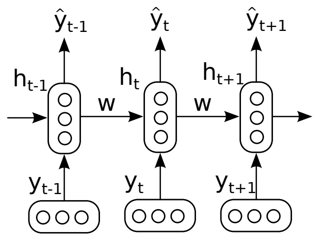


Figure 1.1: Graphical representation of the recurrent neural network from Equation 1.2. w represents the parameters of the network.

encoding $\boldsymbol{y}$ into vector representation. The *states* of the RNN can be used to make predictions about the most likely next element of a sequence.

Given a sequence $\boldsymbol{y} = \{y_1, ..., y_i, ..., y_t\}$ of t words, an RNN can be trained to estimate the probability distribution for the sequence $p(\boldsymbol{y}) = \prod_{i=1}^{t} p(y_i|\{y_1, ..., y_{i-1}\})$ through a recursively-defined function, which encodes the sequence one element at a time:

$$h_i = f(h_{i-1}, y_i, \theta) \tag{1.2}$$

One possible way to implement f (see Figure 1.1) is:

$$f(h_i, y_i) = \sigma(W_h \cdot h_{i-1} + W_w \cdot y_i) \tag{1.3}$$
$$\hat{y}_i = SoftMax(W_{\hat{y}} \cdot h_i) \tag{1.4}$$

Where h_i is the hidden state of the RNN, which contains information about the whole input sequence up to step i, with a focus on the current word. $W_h \in \mathbb{R}^{n \times n}$ is a transition matrix for the next state, n is the number of hidden units, $W_w \in \mathbb{R}^{n \times d}$ is a weight matrix that transforms a new input word embedding $y_i \in \mathbb{R}^d$, σ is a non-linearity such as sigmoid or hyperbolic tangent, and $W_{\hat{y}} \in \mathbb{R}^{K \times n}$ transforms h_i to the vocabulary size K. $\hat{y}_i$ is the estimated probability of the next word, which is computed after applying the softmax function. Since the parameters W_h,W_w and $W_{\hat{y}}$ are shared across the recursive computation, we can apply this model to an input sequence of arbitrary size.

The parameters of an RNN are typically trained with gradient descent using the *Back-propagation through time (BPTT)* algorithm (Werbos, 1990). Frameworks for symbolic computation, such as PyTorch[6], greatly simplify the development and training of modern deep learning models, through providing automatic computation of derivatives over the loss function.

[6] `https://pytorch.org/`

1.3.3 Structured prediction

Many problems in Machine Learning and NLP require the prediction of *sequences* of classes (e.g., the translation of a sentence from English to German), rather than a single class (e.g., whether a paragraph's sentiment is positive or negative). This class of problems is called **structured prediction**, part of which is automatic summarization.

Following (Bahdanau et al., 2015; Rush et al., 2015), we can define automatic abstractive summarization in a probabilistic, structured prediction setting. Given an input sequence of T_x words $\boldsymbol{x} = \{x_1, ..., x_{T_x}\}$ coming from a fixed-length input vocabulary V_x of size K_x, the goal of summarization is to produce a condensed sequence of T_y summary words $\boldsymbol{y} = \{y_1, ..., y_{T_y}\}$ from a summarization vocabulary V_y of size K_y (where $T_x \gg T_y$).

This is a structured prediction problem, which can be estimated by learning a probabilistic mapping for the summary $\boldsymbol{y}$, given the input sequence $\boldsymbol{x}$ (Dietterich et al., 2008; Goodfellow et al., 2016), which can be decomposed as:

$$\arg\max_{\boldsymbol{y}} p(\boldsymbol{y}|\boldsymbol{x}, \theta) = \arg\max_{\boldsymbol{y}} \prod_{i}^{T_y} p(y_i|\{y_0, ..., y_{i-1}\}, \boldsymbol{x}, \theta) \tag{1.5}$$

The mapping composes of a rich internal structure over the input data, and is further parameterized by model parameters θ. Further note that this mapping is a conditional language model which considers the full previous generation history. Modern machine learning techniques can be utilized to learn such a mapping in a data-driven manner, using a large collection of input-summary pairs (Rush et al., 2015).

Under our definition, summarization is *abstractive*, since a system generating $\mathbf{y}$ will consider all words from V_y when producing its output, and condition on the input sequence. In contrast, under this definition, an *extractive* system would model $p(\boldsymbol{z}|\boldsymbol{x})$, where $\boldsymbol{z}$ is coming from a limited vocabulary consisting of only the words contained in the input ($V_z = \boldsymbol{x}_{[1,...,T_x]}$).

Encoder-decoder framework

The **Encoder-Decoder** is a general framework (Cho et al., 2015) for sequence prediction, in which the distribution $p(y|x)$ is learned using two neural network components: an *encoder* and a *decoder*. The two are jointly trained in an end-to-end fashion, to minimize a loss function such as the negative log likelihood, using a data set of N pairs of input and output sequences:

$$\mathcal{L} = -\frac{1}{N} \sum_{k=1}^{N} \sum_{i=1}^{T_y} log(p(y_i^{(k)}|\{y_0^{(k)}, ..., y_{i-1}^{(k)}\}, \mathbf{x}^{(k)})) \tag{1.6}$$

The general definition of the framework with respect to temporal data, following (Bahdanau et al., 2015), is as follows (also visually represented in

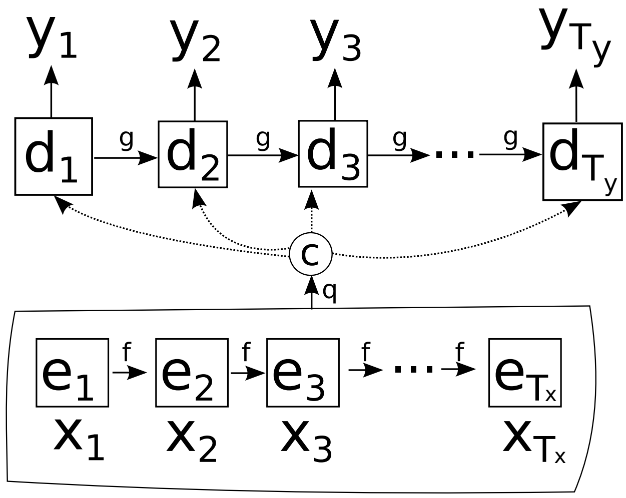


Figure 1.2: Visual representation of the encoder-decoder framework (redrawn from (Cho et al., 2014)).

Figure 1.2). An *encoder* component e takes the input sequence $\boldsymbol{x}$ one element at a time, to produce a sequence of order-sensitive *annotations* (or representations) e_i, using a function f:

$$e_i = f(x_i, e_{i-1}) \tag{1.7}$$

$$c = q(e_1, ..., e_{T_x}) \tag{1.8}$$

A function q is then used to generate the so-called *context* c, which compresses the representations, or selects a subset. A *decoder* component, in turn, generates the output sequence y one element at a time, by decomposing $p(y|x)$ and conditioning on the context:

$$p(\boldsymbol{y}) = \prod_{i=1}^{T_y} p(y_i|\{y_1, ...y_{i-1}\}, c) \tag{1.9}$$

$$p(y_i|\{y_1, ...y_{i-1}\}, c) = g(y_{i-1}, d_i, c) \tag{1.10}$$

Where d_i is an annotation produced by the *decoder* at step i, representing the decoding history. The process finishes when a certain pre-defined sequence limit is reached, or when a special End-of-sequence (EOS) symbol is emitted by the system.

The encoder can be interpreted as producing a lower-dimensional representation, or rather, a sequence of representations, which encapsulate key properties of the input. This representation is then used by the decoder to increase the dimensionality of the signal, and produce the output sequence. In textual prediction, the decoder can also be interpreted as a language model, which uses the context to learn to generate task-specific and meaningful sequences of words, taking into account its generation history.

Originally, the encoder-decoder approach was implemented using recurrent neural networks Sutskever et al. (2014); Cho et al. (2015); Bahdanau et al. (2015) (Section 1.3.2). Later research proposed alternative architectures using convolutional layers Gehring et al. (2017) or only using so-called attention layers Vaswani et al. (2017). In this thesis, we mainly utilize recurrent neural networks.

Decoding using Beam Search

During inference, we can use our trained encoder-decoder model to generate a summary $\boldsymbol{y}^*$ for a new input article following Equation 3.3:

$$\boldsymbol{y}^* = \arg\max_{\boldsymbol{y}} p(\boldsymbol{y}|\boldsymbol{x}, \theta) = \arg\max_{\boldsymbol{y}} \prod_{i}^{T_y} p(y_i|\{y_0, ..., y_{i-1}\}, \boldsymbol{x}, \theta) \tag{1.11}$$

Since an exact computation of the *argmax* is expensive over a potentially large vocabulary containing thousands of words, the decoding is often approximated using a beam search algorithm. Beam search limits decoding to K potential hypotheses[7] at each position of the summary, with the most likely hypothesis returned in the end. Algorithm 1 in Rush et al. (2015) contains an example implementation of beam search.

1.3.4 Low-resource structured prediction

Training an encoder-decoder summarization model following Equation 1.6 requires having access to a large training dataset of articles with example summaries (commonly referred to as a *parallel dataset*). In practice, we may only have very few such examples, in particular when focusing on a low-resource domain, such as summarization of scientific articles to generate press releases (Chapter 4.2). This over-reliance on parallel data has spurred recent work that focuses on *lower-resource approaches to structured prediction*.

Large-scale alignment

For many text-to-text rewriting tasks, such as sentence compression or machine translation, millions of example training pairs might occur naturally within existing textual resources available on the internet, such as in Wikipedia articles or blog posts. The idea of **large-scale alignment** is to mine for and extract such examples programmatically, using an alignment

[7] Values between 5 and 20 often work well in practice.

algorithm. The datasets produced through large-scale alignment are typically referred to as *pseudo-parallel datasets*, to distinguish them from parallel datasets that contain manually constructed training pairs.

The majority of previous work on large-scale alignment is in machine translation, where adding pseudo-parallel pairs to an existing parallel dataset has been shown to boost the translation performance. For example, for translation from German to English, an alignment algorithm can be used to analyze large collections of raw German and English articles, extracting German-English sentence pairs that have a similar meaning. Alignment for translation is often implemented using a sentence-level similarity function (Munteanu and Marcu, 2005; Uszkoreit et al., 2010), for example, by computing the lexical overlap between the original English sentences and an initial English translation of the German sentences. The initial translation can be obtained, for example, through simple word replacement with a German-English dictionary.

More recently, text embeddings (Section 1.3.1) have been gaining popularity as a similarity function for large-scale alignment. (Marie and Fujita, 2017) train two separate word embedding models on two monolingual corpora, and then project the embeddings into the same space. They construct sentence embeddings for each sentence by averaging its constituent word embeddings. To extract their pseudo-parallel corpora, they compute cosine similarity across all possible sentence pairs in their datasets.

When it comes to *monolingual tasks*, such as automatic text simplification, there has been little previous work using large-scale alignment methods. (Barzilay and Elhadad, 2003) develop a hierarchical alignment approach of first clustering paragraphs on similar topics before performing alignment on the sentence level. They argue that, for monolingual data, the pre-clustering of larger textual units is more robust to noise compared to fine-grained sentence matching applied directly on the dataset level. In Chapter 4 of this thesis, we propose a new method for large-scale alignment, which focuses specifically on monolingual tasks.

Backtranslation

Backtranslation (Sennrich et al., 2016a; Edunov et al., 2018) is a simple, yet effective technique for generating synthetic training data for a structured prediction task. The synthetic data can be used to either augment an existing parallel dataset or on its own.

Backtranslation assumes *some* parallel training data for the target task is already available, and that there is *additional non-parallel data*, such as raw articles, that can be exploited. For example, if the target task is translation from German to English (DE $\rightarrow$ EN), we may have some parallel DE-EN sentence pairs, as well as a lot of auxiliary English sentences without any German sentences to match. Using the parallel DE-EN data we do have available, we can train an intermediate model on the *reverse task* (EN $\rightarrow$ DE). This model can then be applied to each auxiliary English sentence we have available to generate synthetic German sentences. The extra sentence pairs can be used to augment the original German-English training dataset, expanding its capacity. Although the synthetic pairs can contain some noise, they may provide valuable training signal to the final DE $\rightarrow$ EN translation model, which resulting in better overall performance. For a more in-depth discussion of backtranslation in different scenarios, please refer to Edunov et al. (2018).

Backtranslation is especially useful for low-resource language pairs, and can even be applied to build fully unsupervised machine translation systems (Lample et al., 2018). Recent work has also successfully applied backtranslation to monolingual tasks, such as text style transfer (Prabhumoye et al., 2018; Lample et al., 2019).

Chapter 2

Overview of the contributions of this thesis

This thesis is cumulative and consists of four open-access publications:

1. *Chapter 3.1*: Nikolov, N. I., Pfeiffer, M., and Hahnloser, R. (2018c). Data-driven summarization of scientific articles. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA)
2. *Chapter 3.2*: Nikolov, N. I., Calmanovici, A., and Hahnloser, R. (2019). Summary refinement through denoising. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2019*
3. *Chapter 4.1*: Nikolov, N. I. and Hahnloser, R. (2019). Large-scale hierarchical alignment for data-driven text rewriting. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2019*
4. *Chapter 4.2*: Nikolov, N. I. and Hahnloser, R. (2020). Abstractive document summarization without parallel data. In *Proceedings of the Twelfth International Conference on Language Resources and Evaluation (LREC 2020)*, Paris, France. European Language Resources Association (ELRA)

The text of the publications is reprinted without any changes to their original content.

In the publications, we consider several problems in data-driven abstractive document summarization. The problems fall into two broad categories: **high-resource summarization** and **low-resource summarization**. Below, we give an overview of each of the problems, while Chapter 5 contains further discussion and suggestions for future work.

2.1 High-resource summarization

2.1.1 Data-driven Summarization of Scientific Articles

Problem and motivation

We started the work that lead to Chapter 3 by identifying two important challenges in current text summarization research. First, there are very few existing large-scale datasets that target abstractive summarization using long input articles and output summaries, such as paragraphs or whole documents. Abstractive summarization research has predominantly focused on the generation of single-sentence summaries (Rush et al., 2015), or summaries which consist of 3 to 5 sentences (See et al., 2017a). Second, most existing summarization datasets target the news domain. Ideally, a summarization system should be tested on datasets from multiple domains, to ensure it is consistently performing well, as well as on datasets which contain very long articles and summaries, to ensure the approach is scalable.

With these challenges in mind, we realized that one unexploited resource that can be used for summarization is **scientific journal articles**. All scientific papers come with a long, carefully constructed abstractive summary, in the form of the abstract of the article, as well as with an even more compressed summary, in the form of the title. Furthermore, large repositories that contain scientific papers are readily available: for example, PubMed[1] contains the full text of over 1.5 million open access papers in the biomedical domain, as well as over 20 million abstracts and titles of closed-access papers.

Overview of our contribution

In our work, we utilize such open-access resources to create two novel challenging large-scale datasets for **scientific article summarization**: generating the title of a paper from its abstract (*title-gen* task), and the abstract from the full body (*abstract-gen* task). The *abstract-gen* task is particularly challenging, with each scientific article containing 172 sentences on average, and each abstract containing 14 sentences on average, which is much larger than the 3 to 5 sentence summaries present in the CNN/DailyMail dataset (See et al., 2017a), for example. At the time our paper was written, no prior work had considered testing data-driven abstractive models on such challenging benchmarks. Section 3.1.3 contains a more in-depth description and statistics on the datasets.

In Section 3.1.5, we test a wide range of existing data-driven summarization approaches on our novel datasets. We aim to examine to what extent state-of-the-art extractive and abstractive summarization models are fit for

[1] `https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/`

these tasks. We were also interested in investigating how well these models, which are typically exclusively tested on news articles, work in the scientific domain, which comprises a complex writing structure, long sentences, as well as a complex scientific jargon.

Our analysis shows that all models that we tested perform well on the *title-gen* task, and are capable of generating realistic-looking titles of papers, with a few potential areas for improvement. However, most models struggled with the more challenging *abstract-gen* task, with many of the synthetic abstracts lacking a coherent structure and containing many repetitions. Our work highlights that much further research is still needed to scale sequence models to very large sequences. Our benchmark datasets could prove to be useful testbeds for novel architectures that aim to address this scalability problem.

2.1.2 Summary Refinement through Denoising

Problem and motivation

Although automatic summarization research has seen huge advancements in the last decade, the vast majority of current summarization models still produce a large number of errors during generation See et al. (2017a). These errors inspired us to focus on developing general methods for improving the quality of summarization outputs in Section 3.2. One common problem plaguing many text generation systems, which we focus on in particular, is the unnecessary repetition of information.

Previous work that aims to control information repetition in text generation models tackles the problem on the modeling side, by introducing novel components in the encoder-decoder framework that keep track of the generation history to detect repetitions of words or phrases (Tu et al., 2016; See et al., 2017a). Alternatively, repetitions can also be detected during inference, through eliminating Beam hypotheses (see Section 1.3.3) that contain high levels of n-gram repetitions (Chen and Bansal, 2018a).

In our work, we instead treat information repetition as a post-processing problem that is tackled by an auxiliary sequence model. The auxiliary model is trained to denoise machine-generated summaries to improve their quality. We call our approach **summary refinement**.

Our motivation for focusing on this direction is three-fold:

1. Such a post-processing method applies to any summarization system, irrespective of whether it is extractive or abstractive. Thus, summary refinement is a general framework for improving the quality of the generated text, rather than a specific mechanism relevant only for a specific modeling paradigm, such as for recurrent neural networks.

2. The post-processing framework can be extended to target not only information repetition, but also other types of errors produced by text generation systems, such as lack of coherence in the generated text, or grammatical errors. Furthermore, our approach is potentially applicable to noisy human text as well, similar to work in automatic error correction (Xie et al., 2018a).

3. In Figure 3.6, we show that even state-of-the-art abstractive systems that already have some built-in mechanisms for reducing redundancy still produce abstractive summaries that contain significant levels of redundancy. Therefore, there is still room for improvement over existing approaches for controlling redundancy.

Overview of our contribution

Our approach to summary denoising consists of two steps (see Figure 3.4 and Section 3.2.3). First, we apply noise functions to perturb a dataset of existing clean summaries, to generate parallel datasets of pairs of clean and noisy summaries. Second, we train standard sequence-to-sequence models to generate clean from noisy summaries.

In our work, we experiment with denoising summaries produced by a wide range of existing extractive and abstractive summarization systems to investigate in which cases our approach might be most beneficial. Furthermore, we also test three different noise types (Section 3.2.3) that introduce diverse repetition errors in the summaries.

Our analysis demonstrates that our method is effective at reducing information repetition in the summaries. For extractive summarization (Section 3.2.5), applying our method even leads to improved performance in terms of ROUGE. For abstractive summarization (Section 3.2.5), although applying our method also leads to a reduction in redundancy, this can also result in a decrease in ROUGE due to deletion of relevant content from the summaries.

2.2 Low-resource summarization

2.2.1 Large-scale Hierarchical Alignment

Problem and motivation

Data-driven summarization methods work best when trained on high-quality parallel datasets consisting of paired articles with summaries. As discussed in Section 1.2.3, the availability of such datasets is limited. This lack of parallel data prohibits the use of summarization systems in diverse domains where they might be needed. For example, currently, there is no parallel dataset for the task of summarizing scientific articles to generate press releases.

With this motivation in mind, in Chapter 4, we switch our focus from high-resource approaches to summarization to low-resource summarization. Our focus is on developing methods that do not rely on parallel data, but can instead be trained using *psuedo-parallel data* extracted from raw corpora (see Section 1.3.4). Our motivation for focusing on extraction of pseudo-parallel data is three-fold:

1. In many domains, there are vast amounts of raw data that can be exploited to construct pseudo-parallel datasets for text-to-text rewriting tasks. For example, there are open-access repositories that contain millions of scientific papers and press releases, with limited links between them (see Section 4.2.6).

2. The alternative to constructing pseudo-parallel datasets is to use machine learning methods which aim to match the distribution of two text collections as a whole, without relying on any labeled examples. For example, (Fu et al., 2017; Shen et al., 2017; Yang et al., 2018) propose neural architectures for text style transfer that can learn disentangled latent representations of content separate from stylistic attributes present in the datasets, such as gender or political inclinations. These methods can potentially be applied to more structured tasks, such as sentence paraphrasing or text simplification. However, they currently face several challenges, such as low meaning preservation from the input to the output[2]. Furthermore, most of these systems are non-differentiable; thus, they are less straightforward to use in practice. We, therefore, do not consider any such systems in this thesis. Shen et al. (2017) provide a good introduction on these methods.

3. The majority of previous work on pseudo-parallel dataset construction focuses on machine translation (see Section 4.1.2), where the approach has been successful. There is, however, little work that uses alignment to tackle monolingual text-to-text rewriting tasks, such as text style transfer. This makes alignment for monolingual tasks an area that is worthy of further research.

Overview of our contribution

In Section 4.1, we consider the problem of large-scale alignment for monolingual tasks. We propose a novel hierarchical alignment method (Figure 4.1, Section 4.1.3) based on nearest neighbor search of document and sentence embeddings. Given two raw datasets that contain comparable documents

[2] We performed some preliminary experiments using the system from Shen et al. (2017), which performed very poorly on the text simplification task from Section 4.1.5.

written in two text styles, the method extracts pseudo-parallel monolingual training pairs through two steps: *document alignment*, which extracts pairs of documents that are similar in meaning; and *sentence alignment*, which extracts suitable sentence pairs.

We perform extensive automatic and human evaluations of our method. First, we evaluate a range of different embedding approaches using a labeled corpus focusing on the task of automatic text simplification (Section 4.1.4). Second, we perform an extrinsic evaluation of our method on the task of text simplification from the normal Wikipedia to the Simple Wikipedia (Section 4.1.5). Third, we also perform an evaluation using human raters (Section 4.1.5). Our results demonstrate that our method is capable of extracting pseudo-parallel sentence pairs that are useful for augmenting existing parallel datasets, as well as for tackling novel text-to-text rewriting tasks.

2.2.2 Abstractive Document Summarization without Parallel Data

Problem and motivation

Developing abstractive document summarization methods that require fewer or no parallel training examples would increase the potential impact of summarization systems for under-resourced languages, as well as in the diverse domains for which parallel data is limited. For example, they could be used for automatically generating press releases from scientific articles, which could make science more accessible to non-experts (see Section 1.2.5).

With these motivations in mind, in Chapter 4.2, we develop a novel abstractive document summarization system that does not rely on any parallel data, but can instead be trained using pseudo-parallel resources *only*. We choose to build on our large-scale alignment work from the previous section because it provides a simple method for training an abstractive sentence paraphrasing model using raw text collections.

Overview of our contribution

We propose a novel approach to non-parallel summarization which consists of two consecutive steps (see Figures 4.2 and 4.3, Section 4.2.3). First, given an input article to be summarized, an unsupervised *sentence extractor* is used to select salient sentences from the article. Second, a *sentence abstractor* paraphrases each of the extracted sentences to generate the final abstractive summary. The sentence abstractor is trained on pseudo-parallel data extracted using our large-scale alignment method, as well as on additional data generated through backtranslation (see Section 1.3.4). The two components are combined in a pipeline which performs extraction followed by abstraction.

We perform experiments on two tasks. First, we compare the performance of our summarization pipeline to models trained on parallel data, using the

CNN/DailyMail news article summarization benchmark (Section 4.2.5). We aim to study the performance gap between our summarization pipeline and models trained on parallel data, to quantify to what extent our approach could be a feasible substitute. We show that there is a significant gap between parallel and pseudo-parallel models, which is expected due to the difficulty of the task. The gap can be decreased by training the models on additional synthetic data generated through backtranslation; however, this is still insufficient to reach the performance of the parallel models. Furthermore, we show that our alignment method can reliably reconstruct the CNN/DailyMail parallel dataset. This result indicates that our summarization approach should work particularly well in domains for which high-quality raw data is available.

We also successfully apply our system to the novel task of automatically generating a press release for a scientific journal article (Section 4.2.6). As mentioned earlier, this task is particularly suitable for our system, because there are large quantities of scientific papers and press releases, without many links between them. Our abstractive method outperforms several extractive baselines on this task, demonstrating that the abstractive paraphrasing step learned from pseudo-parallel data is beneficial.

Chapter 3

Data-driven Summarization and Summary Refinement

3.1 Data-driven Summarization of Scientific Articles

Originally published as: Nikolov, N. I., Pfeiffer, M., and Hahnloser, R. (2018c). Data-driven summarization of scientific articles. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA)

Abstract

Data-driven approaches to sequence-to-sequence modelling have been successfully applied to short text summarization of news articles. Such models are typically trained on input-summary pairs consisting of only a single or a few sentences, partially due to limited availability of multi-sentence training data. Here, we propose to use scientific articles as a new milestone for text summarization: large-scale training data come almost for free with two types of high-quality summaries at different levels - the title and the abstract. We generate two novel multi-sentence summarization datasets from scientific articles and test the suitability of a wide range of existing extractive and abstractive neural network-based summarization approaches. Our analysis demonstrates that scientific papers are suitable for data-driven text summarization. Our results could serve as valuable benchmarks for scaling sequence-to-sequence models to very long sequences.

3.1.1 Introduction

The goal of automatic text summarization is to produce a shorter, informative version of an input text. While extractive summarization only consists of selecting important sentences from the input, abstractive summarization *generates* content without explicitly re-using whole sentences (Nenkova et al., 2011). Text summarization is an area with much promise in today's age of information overflow. In the domain of scientific literature, the rate of publications grows exponentially (Hunter and Cohen, 2006), which calls for efficient automatic summarization tools.

Recent state-of-the-art summarization methods learn to summarize in a data-driven way, relying on large collections of input-summary training examples. The majority of previous work focused on short summarization of news articles, such as to generate a title (Rush et al., 2015; Nallapati et al., 2016). One major challenge is to scale these methods to process long input/output sequence pairs. Currently, availability of large-scale high-quality training data is scarce.

In this paper, we explore the suitability of **scientific journal articles** as a new benchmark for data-driven text summarization. The typical well-structured format of scientific papers makes them an interesting challenge, and provides plenty of freely available training data, because every article comes with a summary in the form of its abstract, and, in even more compressed form, its title. We make a fist step towards summarization of whole scientific articles, by composing two novel large datasets for **scientific summarization**: title-abstract pairs (*title-gen*), composed of 5 million papers in the biomedical domain, and abstract-body pairs (*abstract-gen*) composed of 900k papers[1]. The second dataset is particularly challenging, because it is intended for summarizing the full body of the paper in terms of the abstract (the lengths of input/output sequences are substantially longer than what has been considered so far in previous research, see Table 3.1).

We evaluate a range of existing state-of-the-art approaches on these datasets: extractive approaches based on word embeddings, as well as word, subword, and character-level encoder-decoder models that use recurrent as well as convolutional modules. We perform a quantitative and qualitative analysis of the models' outputs.

3.1.2 Background

Extractive Summarization

Given an input document consisting of T_s sentences $\boldsymbol{s} = \{s_1, ..., s_{T_s}\}$, the goal of **extractive summarization** is to select the K most salient sentences as the output summary. Extractive summarization typically involves a **sentence representation module** e, that represents each input sentence s_i in a common space as $r_i = e(s_i)$, e.g. as a vector of real numbers; as well as a **ranking module** *score*, that weights the salience $w_i = score(r_i)$ of each

[1] Both datasets are available at `https://github.com/ninikolov/data-driven-summarization`, including versions with and without preprocessing.

sentence. A typical approach to unsupervised extractive summarization is to implement w_i as the similarity between r_i and a document representation (or a **document centroid**) $r_d = e(d)$ (Radev et al., 2004). Alternatively, one can compute w_i as the **sentence centrality**, which is an adjacency-based measure of sentence importance (Erkan and Radev, 2004).

In this work, we propose two simple unsupervised baselines for extractive summarization, both of which rely on word embeddings (Mikolov et al., 2013). The first, *tfidf-emb*, represents each sentence in the input document as the weighted sum of its constituent word embeddings, similar to (Rossiello et al., 2017):

$$r_i = e(s_i) = \frac{1}{t_i} \sum_{x \in s_i} t(x) \cdot E(x), \tag{3.1}$$

where $E(x)$ is the embedding of word x, $t(x)$ is an (optional) weighting function that weighs the importance of a word, and $t_i = \sum_{x \in s_i} t(x)$ is a normalization factor. As a weighing function, we use the term-frequency inverse document frequency (TF-IDF) score, similar to (Brokos et al., 2016). Each sentence embedding r_i can then be ranked by computing its cosine similarity $sim(r_d, r_i)$ to a document centroid r_d, computed similarly as r_i. The summary consists of the top K sentences with embeddings most similar to the document embedding.

The second baseline, *rwmd-rank*, ranks the salience of a sentence in terms of its similarity to all the other sentences in the document. All similarities are stored in an intra-sentence similarity matrix W. We use the Relaxed Word Mover's Distance (RWMD) to compute this matrix (Kusner et al., 2015):

$$\begin{aligned} W_{ij} = rwmd(s_i, s_j) &= \max(rwmd_p(s_i, s_j), rwmd_p(s_j, s_i)) \\ rwmd_p(s_i, s_j) &= \sum_{x \in s_i} \min_{x' \in s_j} dist(E(x), E(x')), \end{aligned} \tag{3.2}$$

where s_i and s_j are two sentences and $dist(E(x), E(x'))$ is the Euclidean distance between the embeddings of words x and x' in the sentences. To rank the sentences, we apply the graph-based method from the LexRank system (Erkan and Radev, 2004). LexRank represents the input as a highly connected graph, in which vertices represent sentences, and edges between sentences are assigned weights equal to their similarity from W. The centrality of a sentence is then computed using the PageRank algorithm (Page et al., 1999).

Abstractive Summarization

Given an input sequence of T_x words $\boldsymbol{x} = \{x_1, ..., x_{T_x}\}$ coming from a fixed-length input vocabulary V_x of size K_x, the goal of abstractive summarization is to produce a condensed sequence of T_y summary words $\boldsymbol{y} = \{y_1, ..., y_{T_y}\}$ from a summarization vocabulary V_y of size K_y, where $T_x \gg T_y$. Abstractive summarization is a structured prediction problem that can be solved by learning a probabilistic mapping $p(\boldsymbol{y}|\boldsymbol{x}, \theta)$ for the summary $\boldsymbol{y}$, given the input sequence $\boldsymbol{x}$ (Dietterich et al., 2008):

$$p(\boldsymbol{y}|\boldsymbol{x}, \theta) = \prod_{i}^{T_y} p(y_i|\{y_0, ..., y_{i-1}\}, \boldsymbol{x}, \theta). \tag{3.3}$$

The **encoder-decoder architecture** is a recently proposed general framework for structured prediction (Cho et al., 2015), in which the distribution $\arg\max_{\boldsymbol{y}} p(\boldsymbol{y}|\boldsymbol{x}, \theta)$ is learned using two neural networks: an **encoder** network e, which produces intermediate representations of the input, and a **decoder** language modelling network d, which generates the target summary. The decoder is conditioned on a context vector c, which is recomputed from the encoded representation at each decoding step. The encoder-decoder was first implemented using Recurrent Neural Networks (RNNs) (Sutskever et al., 2014; Cho et al., 2014) that process the input sequentially. Recent studies have shown that convolutional neural networks (CNNs) (LeCun et al., 1998) can outperform RNNs in sequence transduction tasks (Kalchbrenner et al., 2016; Gehring et al., 2017). Unlike RNNs, CNNs can be efficiently implemented on parallel GPU hardware. This advantage is particularly important when working with very long input and output sequences, such as whole paragraphs or documents. CNNs create hierarchical representations over the input in which lower layers operate on nearby elements and higher layers implement increasing levels of abstraction.

In this work, we investigate the performance of three existing systems that operate on different levels of sequence granularity. The first, *lstm*, is a recurrent Long Short Term Memory (LSTM) encoder-decoder model (Sutskever et al., 2014) with an attention mechanism (Bahdanau et al., 2015) that operates on the word level, processing the input sequentially. The second system, *fconv*, is a convolutional encoder-decoder model from (Gehring et al., 2017). *fconv* works on the subword level and segments words into smaller units using the byte pair encoding scheme. Using subword units improves the generation quality when dealing with rare or unknown words (Sennrich et al., 2016c). The third system, *c2c*, is a character-level encoder-decoder model from (Lee et al., 2017) that models $\boldsymbol{x}$ and $\boldsymbol{y}$ as individual characters, with no explicit segmentation between tokens. *c2c* first builds representations of groups of characters in the input using a series of convolutional layers. It then applies a recurrent encoder-decoder, similar to the *lstm* system.

Scientific Articles

Previous research on summarization of scientific articles has focused almost exclusively on extractive methods (Nenkova et al., 2011). In (Lloret et al., 2013), the authors develop an unsupervised system for abstract generation of biomedical papers, that first selects relevant content from the body, following which it performs an abstractive information fusion step. More recently, (Kim et al., 2016) consider the problem of supervised generation of sentence-level summaries for each paragraph of the introduction of a paper. They construct a training dataset of computer science papers from arXiv, selecting the most informative sentence as the summary of each paragraph using the Jaccard similarity. Thus, their target summary is fully contained in the input. In (Collins et al., 2017), they develop a supervised extractive summarization framework which they apply to a dataset of 10*k* computer science papers. To the best of our knowledge, our work is the first on abstractive title generation of scientific articles, and is the first to consider supervised generation of the absctract directly from the full body of the paper. The datasets we utilize here are also substantially larger than in previous work on scientific summarization.

Scientific articles are potentially more challenging to summarize than news articles because of their compact, inexplicit discourse style (Biber and Gray, 2010). While the events described by news headlines frequently recur in related articles, a scientific title focuses on the unique contribution that sets a paper apart from previous research (Teufel and Moens, 2002). Furthermore, while the first two sentences of a news article are typically sufficiently informative to generate its headline (Nallapati et al., 2016; Teufel and Moens, 2002), the first sentences of the abstract or introduction of a paper typically contain background information on the research topic. Constructing a good scientific title thus requires understanding and integrating concepts from multiple sentences of the abstract.

3.1.3 Datasets

To investigate the performance of encoder-decoder neural networks as generative models of scientific text, we constructed two novel datasets for scientific summarization. For *title-gen* we used MEDLINE[2], whereas for *abstract-gen* we used the PubMed open access subset[3]. MEDLINE contains scientific metadata in XML format of $\sim$ 25 million papers in the biomedical domain,

[2] https://nlm.nih.gov/databases/download/pubmed_medline.html
[3] https://ncbi.nlm.nih.gov/pmc/tools/openftlist

Table 3.1: Statistics (mean and standard deviation) of the two scientific summarization datasets: *title-gen* and *abstract-gen*. Token/sentence counts are computed with NLTK.

title-gen	*Abstract*	*Title*
Token count	245 ± 54	15 ± 4
Sentence count	14 ± 4	1
Sent. token count	26 ± 14	-
Overlap	$73\% \pm 18\%$	
Repeat	$44\% \pm 11\%$	-
Size (tr/val/test)	$5'000'000/6844/6935$	

abstract-gen	*Body*	*Abstract*
Token count	4600 ± 1987	254 ± 54
Sentence count	172 ± 78	10 ± 3
Sent. token count	26 ± 17	26 ± 14
Overlap	$68\% \pm 10\%$	
Repeat	$74\% \pm 7\%$	$44\% \pm 11\%$
Size (tr/val/test)	$893'835/10'916/10'812$	

whereas the PubMed open access subset contains metadata and full text of $\sim$ 1.5 million papers.

We processed the XML files[4] to pair the abstract of a paper to its title (*title-gen* dataset) or the full body (*abstract-gen*), skipping any figures, tables or section headings in the body. We then apply several preprocessing steps from the MOSES statistical machine translation pipeline[5], including tokenization and conversion to lowercase. Any URLs were removed, all numbers replaced with #, and any pairs with abstract lengths not in the range of 150-370 tokens, title lengths not within 6-25 tokens, and body lengths not within 700-10000 tokens were excluded.

The *Overlap* $o(\mathbf{x}, \mathbf{y}) = \frac{|\{\mathbf{y}\} \cap \{\mathbf{x}\}|}{|\{\mathbf{y}\}|}$ is the fraction of unique output (summary) tokens $\mathbf{y}$ that overlap with an input token $\mathbf{x}$ (excluding punctuation and stop words). As can be seen in Table 3.1, the overlaps are large in our datasets, indicating frequent reuse of words. The *Repeat* $e(\mathbf{s}) = \frac{\sum_i o(\overline{\mathbf{s}_i}, s_i)}{|\mathbf{s}|}$ is the average overlap of each sentence s_i in a text with the remainder of the text (where $\overline{\mathbf{s}_i}$ denotes the complement of sentence s_i). *Repeat* measures the redundancy of content within a text: a high value indicates frequent repetition of content. Whereas in abstracts there are only moderate levels of repetition, in the bodies the repetition rates are much higher, possibly because concepts and ideas are reiterated in multiple sections of the paper.

[4] We use `https://titipata.github.io/pubmed_parser`.

[5] `github.com/moses-smt/mosesdecoder`

3.1.4 Evaluation Set-up

We evaluated the performance of several state-of-the-art approaches on our scientific summarization datasets. The extractive systems we consider are: *lead*, *lexrank*, *tfidf-emb*, and *rwmd-rank*. The *lead* baseline returns the first sentence of the abstract for *title-gen*, or the first 10 sentences of the body for *abstract-gen*. *lexrank* (Erkan and Radev, 2004) is a graph-based centrality approach frequently used as a baseline in the literature. *emb-tfidf* uses sentence embeddings[6] to select the most salient sentences from the input, while *rwmd-rank* uses the Relaxed Word Mover's Distance (as described in Section 3.1.2). *oracle* estimates an upper bound for the extractive summarization task by finding the most similar sentence in the input document for each sentence in the original summary. We use the Relaxed Word Mover's Distance to compute the output of the oracle.

The abstractive systems we consider are: *lstm*, *fconv*, and *c2c*, described in Section 3.1.2. For *lstm*, we set the input/output vocabularies to 80000, use two LSTM layers of 1000 hidden units each, and word embedding of dimension 500 (we found no improvement from additionally increasing the size of this model). For *c2c* and *fconv*, we use the default hyper-parameters that come with the public implementations provided by the authors of the systems. The *title-gen lstm*, *c2c*, and *fconv* were trained for 11, 8, and 20 epochs, respectively, until convergence.

We were unable to train *lstm* and *c2c* on *abstract-gen* because of the very high memory and time requirements associated with the recurrent layers in these models. We found *fconv* to be much more efficient to train, and we succeeded in training a default model for 17 epochs. For *title-gen*, we used beam search with beam size 20, while for *abstract-gen* we found a beam size of 5 to perform better.

3.1.5 Quantitative Evaluation

In Tables 3.2 and 3.3, we evaluate our approaches using the ROUGE metric Lin (2004), which is a recall-based metric frequently used for summarization, and METEOR Denkowski and Lavie (2014), which is a precision-based metric for machine translation. Overlap can be interpreted as the tendency of the model to directly copy input content instead of generating novel correct or incorrect words; whereas Repeat measures a model's tendency to repeat itself, which is a frequent issue with encoder-decoder models Suzuki and Nagata (2017).

On title generation, *rwmd-rank* achieved the best performance in terms of selecting a sentence as the title. In overall, the abstractive systems significantly

[6]We use the best-performing Word2Vec model from (Chiu et al., 2016), which is trained on PubMed and MEDLINE.

Table 3.2: Metric results for the *title-gen* dataset. R-1, R-2, R-L represent the ROUGE-1/2/L metrics.

Model	R-1	R-2	R-L	MET.	Overlap	Tok. count
oracle	0.386	0.184	0.308	0.146	-	29 ± 14
lead-1	0.218	0.061	0.169	0.077	-	28 ± 14
lexrank	0.26	0.089	0.201	0.089	-	32 ± 14
emb-tfidf	0.252	0.081	0.193	0.082	-	35 ± 17
rwmd-rank	0.311	0.13	0.245	0.116	-	28 ± 13
lstm	0.375	0.173	0.329	0.204	78% ± 20%	12 ± 3
c2c	**0.479**	0.264	**0.418**	0.237	93% ± 10%	14 ± 4
fconv	0.463	**0.277**	0.412	**0.27**	95% ± 9%	15 ± 7

Table 3.3: Metric results for the *abstract-gen* dataset. R-1, R-2, R-L represent the ROUGE-1/2/L metrics.

Model	R-1	R-2	R-L	MET.	Overlap	Repeat	Tok. count
oracle	0.558	0.266	0.316	0.214	-	42% ± 10%	327 ± 99
lead-10	0.385	0.111	0.18	0.138	-	20% ± 4%	312 ± 88
lexrank	0.45	**0.163**	0.213	0.157	-	52% ± 10%	404 ± 131
emb-tfidf	0.445	0.159	0.216	0.159	-	52% ± 10%	369 ± 117
rwmd-rank	**0.454**	0.159	**0.216**	0.167	-	50% ± 10%	344 ± 93
fconv	0.354	0.131	0.209	**0.212**	98% ± 2%	52% ± 28%	194 ± 15

outperformed the extractive systems, as well as the extractive oracle. *c2c* and *fconv* performed much better than *lstm*, with a very high rate of overlap. The ROUGE performance of *c2c* and *fconv* is similar, despite the difference of a few R-2 points in favour of *fconv* (that model is evaluated on a subword-level ground truth file, where we observe a slight increase of 1-2 ROUGE points on average due to the conversion).

On abstract generation, the *lead-10* baseline remained tough to beat in terms of ROUGE, and only the extractive systems managed to surpass it by a small margin. All extractive systems achieved similar results, with *rwmd-rank* having a minor edge, while the abstractive *fconv* performed poorly, even though it performed best in terms of METEOR. We observed a much higher repeat rate in the output summaries than the observed 44% average in the original abstracts (Table 3.1). As revealed by the large Repeat standard deviation for *fconv*, some examples are affected by very frequent repetitions.

3.1.6 Qualitative Evaluation

In Table 3.4 and Figure 3.2, we present two shortened inputs from our *title-gen* and *abstract-gen* test sets, along with original and system-generated summaries. In Figure 3.1, we show a histogram of the locations of input sentences, that estimates which locations were most preferred on average when producing a summary.

We observe a large variation in the sentence locations selected by the extractive systems on *title-gen* (Figure 3.1a), with the first sentence having high importance. Based on our inspection, it is rare that a sentence from the abstract will match the title exactly - the title is also typically shorter than an average sentence from the abstract (Table 3.1). A good title seems to require the selection, combination and paraphrasing of suitable parts from multiple sentences, as also shown by the original titles in our examples. Many of the titles generated by the abstractive systems sound faithful, and at first glance can pass for a title of a scientific paper. The abstractive models are good at discerning important from unimportant content in the abstract, at extracting long phrases, or sometimes whole sentences, and at abstractively combining the information to generate a title. *lstm* is more prone to generate novel words, whereas *c2c* and *fconv* mostly rely on direct copying of content from the abstract, as also indicated by their overlap scores.

Closer inspection of the titles reveals occasional subtle mistakes: for example, in the first example in Table 3.4, the *fconv* model incorrectly selected "scopolamine- and cisplatin- induced" which was investigated in the previous work of the authors and is not the main focus of the article. The model also copied the incorrect genus, "mouse" instead of "rat". Sometimes the generated titles sound too general, and fail to communicate the specifics of the paper: in the second example, all models produced "a model of basal ganglia", missing to include the keyword "reinforcement learning": "a model of reinforcement learning in the basal ganglia". These mistakes highlight the complexity of the task, and show that there is still much room for further improvement.

As shown in Figure 3.1b, the introductory and concluding sections are often highly relevant for abstract generation, however relevant content is spread across the entire paper. Interestingly, in the example in Table 3.2, there is a wide range of content that was selected by the extractive systems, with little overlap across systems. For instance, *rwmd-rank* overlaps with *oracle* by 3 sentences, and only by 1 sentence with *emb-tfidf*. The outputs of the abstractive *fconv* system on abstract generation are poor in quality, and many of the generated abstracts lack coherent structure and content flow. There is also frequent repetition of entire sentences, as shown by the last sentences produced by *fconv* in Table 3.2. *fconv* also appears to only use the first 60 sentences of the paper to construct the abstract (Figure 3.1c).

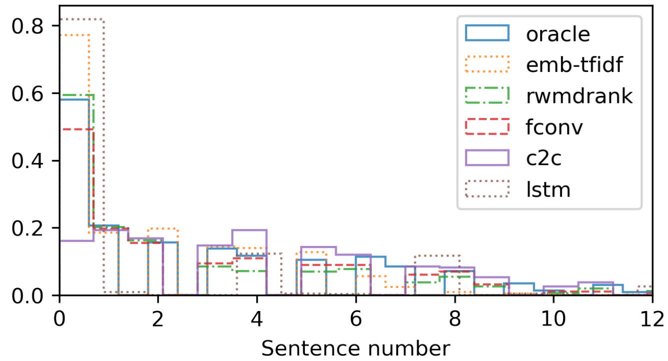
3.1.7 Conclusion

We evaluated a range of extractive and abstractive neural network-based summarization approaches on two novel datasets constructed from scientific journal articles. While the results for title generation are promising, the models struggled with generating the abstract. This difficulty highlights the necessity for developing novel models capable of efficiently dealing with long input and output sequences, while at the same time preserving the quality of generated sentences. We hope that our datasets will promote more work in this area. A direction to explore in future work is hybrid extractive-abstractive end-to-end approaches that jointly select content and then paraphrase it to produce a summary.

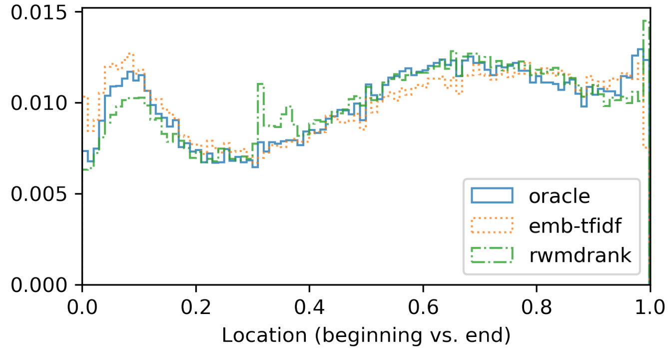
Appendix A

In order to give more insight into our scientific summarization datasets, in Figure 3.3 we plot salient terms obtained by computing Latent Dirichlet Allocation (LDA) topics (Hoffman et al., 2010) on the 5 million abstracts from the *abstract-gen* dataset. We use the LDA implementation of Gensim[7] and set the number of topics to 3, for easier visualization. We use the pyLDAvis library[8] for plotting, displaying the 30 most relevant terms for estimating the LDA topics, as a whole in Figure 3.3a, and as well as for each of the 3 topics individually in Figures 3.3b, 3.3c and 3.3d. The relevance of the terms is computed following Sievert and Shirley (2014).

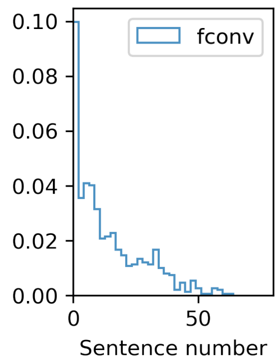
[7] `https://radimrehurek.com/gensim/`
[8] `https://github.com/bmabey/pyLDAvis`.



(a)



(b)



(c)

Figure 3.1: Sentence selection (normalized) histograms computed on the test set, showing the input locations that were most preferred on average by the systems on *title-gen* (a) and *abstract-gen* (b), (c). For (b), we normalize the sentence locations by the length of each paper, to get a better uniform view (there is a large variation in the length of a paper, as shown in Table 3.1). For the abstractive systems, we search for the closest sentences in the input using relaxed word mover's distance (see Section 3.1.2).

Table 3.4: Examples from the test set of *title-gen*. The outputs of the extractive systems are highlighted as: oracle, tfidf-emb, rwmd-rank. For the abstractive systems, we manually highlighted the text of the concepts that are relevant for the task (errors are highlighted in red).

Example 1 Giridharan et al. (2015) Abstract: Amyloid (A)-induced neurotoxicity is a major pathological mechanism of Alzheimer's disease (AD). Our previous studies have demonstrated that schisandrin B (Sch B), an antioxidant lignan from Schisandra chinensis, could protect mouse brain against scopolamine- and cisplatin-induced neuronal dysfunction. In the present study, we examined the protective effect of Sch B against intracerebroventricular (ICV)-infused A-induced neuronal dysfunction in rat cortex and explored the potential mechanism of its action. Our results showed that 26 days co-administration of Sch B significantly improved the behavioral performance of A (1–40)-infused rats in step-through test. At the same time, Sch B attenuated A-induced increases in oxidative and nitrosative stresses *(...)* The aforementioned effects of Sch B suggest its protective role against A-induced neurotoxicity through intervention in the negative cycle of RAGE-mediated A accumulation during AD patho-physiology.
Original title: schisandrin b ameliorates icv-infused amyloid induced oxidative stress and neuronal dysfunction through inhibiting rage / nf-b / mapk and up-regulating hsp / beclin expression
lstm: schisandrin b , an antioxidant lignan from schisandra chinensis , protects against amyloid -induced neurotoxicity
c2c: schisandra chinensis b protects against intracerebroventricular-infused amyloid induced neuronal dysfunction in rat cortex
fconf: schisandrin b protects mouse brain against scopolamine- and cisplatin-induced neurotoxicity in rats
Example 2 Fee (2012) Abstract: In its simplest formulation, reinforcement learning is based on the idea that if an action taken in a particular context is followed by a favorable outcome, then, in the same context, the tendency to produce that action should be strengthened, or reinforced. *(...)* Recent experiments in the songbird suggest that vocal-related BG circuitry receives two functionally distinct excitatory inputs. *(...)* The other is an efference copy of motor commands from a separate cortical brain region that generates vocal variability during learning. Based on these findings, I propose here a general model of vertebrate BG function that combines context information with a distinct motor efference copy signal. *(...)* The model makes testable predictions about the anatomical and functional properties of hypothesized context and efference copy inputs to the striatum from both thalamic and cortical sources.
Original title: oculomotor learning revisited : a model of reinforcement learning in the basal ganglia incorporating an efference copy of motor actions .
lstm: a model of basal ganglia function .
c2c: a general model of vertebrate basal ganglia function .
fconf: a model of basal ganglia function in the songbird .

Example 1 (Pyysalo et al., 2011) Body: **In recent years, there has been a significant shift in focus in biomedical information extraction from simple pairwise relations representing associations such as protein-protein interactions (PPI) toward representations that capture typed, structured associations of arbitrary numbers of entities in specific roles, frequently termed event extraction** [1]. Much of this work draws on the GENIA Event corpus *(...)* This resource served also as the source for the annotations in the first collaborative evaluation of biomedical event extraction methods, the 2009 BioNLP shared task on event extraction (BioNLP ST) [6] as well as for the GENIA subtask of the second task in the series [7, 8]. Another recent trend in the domain is a move toward the application of extraction methods to the full scale of the existing literature, with results for various targets covering the entire PubMed literature database of nearly 20 million citations being made available [9, 10, 11, 12]. **As event extraction methods** initially developed to target the set of events defined in the GENIA / BioNLP ST corpora are now being applied at PubMed scale, it makes sense to ask how much of the full spectrum of gene/protein associations found there they can maximally cover. *(...)* By contrast, we will assume that associations not appearing in this data cannot be extracted: as the overwhelming majority of current event extraction methods **are based on supervised machine learning or hand-crafted rules written with reference to the annotated data**, it reasonable to assume as a first approximation that their coverage of associations not appearing in that data is zero. In this study, we seek to characterize the full range of associations of specific genes/proteins described in the literature and estimate what coverage of these associations event extraction systems relying on currently available resources can maximally achieve. To address these questions, **it is necessary not only to have an inventory of concepts that (largely) covers the ways in which genes/proteins can be associated, but also to be able to estimate the relative frequency with which these concepts are used to express gene/protein associations** in the literature. *(...)* Here, as we are interested in particular in texts describing associations between two or more gene/protein related entities, **we apply a focused selection, picking only those individual sentences in which** two or more mentions co-occur. While this excludes associations in which the entities occur in different sentences, their relative frequency is expected to be low: for example, in the BioNLP ST data, all event participants occurred within a single sentence in 95% of the targeted biomolecular event statements. *(...)* Here, we follow the assumption that when two entities are stated to be associated in some way, the most important words expressing their association will typically be found on the shortest dependency path connecting the two entities (cf. the shortest path hypothesis of Bunescu and Mooney [30]). The specific dependency representation *(...)* Table 3 shows the words most frequently occurring on these paths. This list again suggests an increased focus on words relating to gene/protein associations: expression is the most frequent word on the paths, and binding appears in the top-ranked words. *(...)* Finally, to make this pair data consistent with the TPS event spans, tokenization and other features, we aligned the entity annotations of the two corpora. *(...)* This processing was applied to the BioNLP ST training set, creating a corpus of 6889 entity pairs of which 1119 (16%) were marked as expressing an association (positive). *(...)* Evaluation. We first evaluated each of the word rankings discussed in the section on Identification of Gene/Protein Associations by comparing the ranked lists of words against the set of single words marked as trigger expressions in the BioNLP ST development data. *(...)* To evaluate the capability of the presented approach to identify new expressions of gene/protein associations, we next performed a manual study of candidate words for stating gene/protein associations using the E w ranking. *(...)* We then selected the words ranked highest by E w that were not known, grouped by normalized and lemmatized form, and added for reference examples of frequent shortest dependency paths on which any of these words appear (see example in Table 5). *(...)* If static relations and experimental observations and manipulations are excluded as (arguably) not in scope for event extraction, this estimate suggests that currently available resources for event extraction cover over 90% of all events involving gene/protein entities in PubMed. Discussion. We found that out of all gene/protein associations in PubMed, currently existing resources for event extraction are lacking in coverage of a number of event types such as dissociation, many relatively rare (though biologically important) protein post-translational modifications, as well as some high-level process types involving genes/proteins such as apoptosis. *(...)* This suggests that for practical applications it may be important to consider also this class of associations. *(...)* While these results are highly encouraging, it must be noted that the approach to identifying gene/protein associations considered here is limited in a number of ways: it excludes associations stated across sentence boundaries and ones for which the shortest path hypothesis does not hold, does not treat multi-word expressions as wholes, ignores ambiguity in implicitly assuming a single sense for each word, and only directly includes associations stated between exactly two entities. The approach is also fundamentally limited to associations expressed through specific words and thus blind to e.g. part-of relations implied by statements such as CD14 Sp1-binding site. *(...)* Conclusions. We have presented an approach to discovering expressions of gene/protein associations from PubMed based on named entity co-occurrences, shortest dependency paths and an unlexicalized classifier to identify likely statements of gene/protein associations. Drawing on the automatically created full-PubMed annotations of the Turku PubMed-Scale (TPS) corpus and using the BioNLP09 shared task data to define positive and negative examples of association statements, we distilled an initial set of over 30 million protein mentions into a set of 46,000 unique unlexicalized paths estimated likely to express gene/protein

Figure 3.2: Two examples from the test set of *abstract-gen*. The outputs of the extractive systems are highlighted as: tfidf-emb in green and rwmd-rank in cyan, whereas gray denotes overlap between the two. In **bold** we mark the content that was selected by the *fconv* system (next page in full), and in underline we mark the selection of the *oracle*.

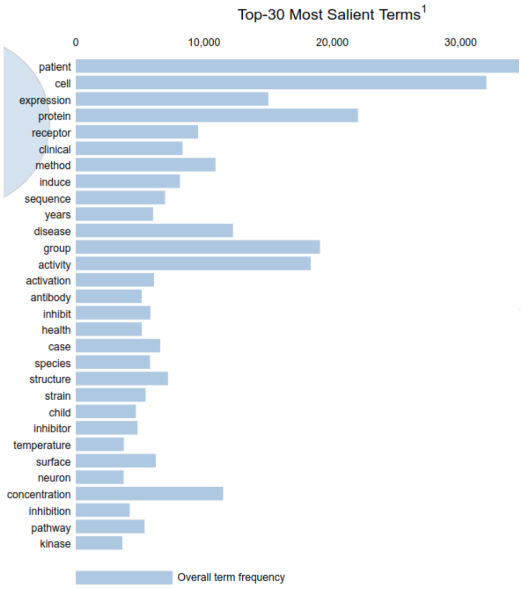
Example 1 Original abstract: Background: Event extraction following the GENIA Event corpus and BioNLP shared task models has been a considerable focus of recent work in biomedical information extraction. This work includes efforts applying event extraction methods to the entire PubMed literature database, far beyond the narrow subdomains of biomedicine for which annotated resources for extraction method development are available. Results: In the present study, our aim is to estimate the coverage of all statements of gene/protein associations in PubMed that existing resources for event extraction can provide. We base our analysis on a recently released corpus automatically annotated for gene/protein entities and syntactic analyses covering the entire PubMed, and use named entity co-occurrence, shortest dependency paths and an unlexicalized classifier to identify likely statements of gene/protein associations. A set of high-frequency/high-likelihood association statements are then manually analyzed with reference to the GENIA ontology. Conclusions: We present a first estimate of the overall coverage of gene/protein associations provided by existing resources for event extraction. Our results suggest that for event-type associations this coverage may be over 90%. We also identify several biologically significant associations of genes and proteins that are not addressed by these resources, suggesting directions for further extension of extraction coverage.

Example 1 fconv: background in recent years , there has been a significant shift in focus in biomedical information extraction from simple pairwise relations representing associations such as protein-protein interactions (ppi) toward representations that capture typed , structured associations of arbitrary numbers of entities in specific roles , frequently termed event extraction . as event extraction methods are based on supervised machine learning or hand-crafted rules written with reference to the annotated data , it is necessary not only to have an inventory of concepts that (largely) covers the ways in which genes / proteins can be associated , but also to be able to estimate the relative frequency with which these concepts are used to express gene / protein associations . results we apply a focused selection , picking king only those individual sentences in which at least one named entity has been tagged . we apply a focused selection , picking only those individual sentences in which at least one named entity has been tagged . we apply a focused selection , picking only those individual sentences in which at least one named entity has been tagged . we apply a probabilistic approach

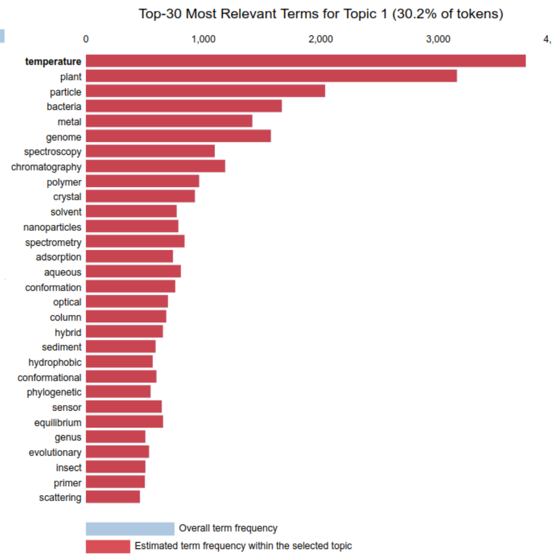
Example 2 (Kumar et al., 2013) Body: **Both alopecia areata (AA) and vitiligo are autoimmune diseases, and their coexistence in the same patient is not uncommon, as vitiligo has been reported to occur in 4.1% of patients of AA and is about 4 times more common in patients with AA than in the general population.** [1] However, their colocalization over the same site is exceedingly rare, with less than five cases being reported in the literature.[2,3,4] **We present a case of a 15-year-old male child who had vitiligo and later developed AA over the** existing lesions of vitiligo over **face and scalp** and have attempted to elucidate the current understanding of mechanisms of coexistence of these two diseases. A 12-year-old boy presented to the skin outpatient department with history of depigmented areas on the scalp, face, neck, arms and legs for 5 years. He also gave a history of development of patchy loss of hair over some of these lesions for 3 years. There was no previous history of any trauma or medications. Family history was not relevant. On examination, there were **depigmented macules** over the **scalp**, **forehead**, **eyebrows**, **eyebrows, perioral, preauricular regions**, neck, elbows, hands, feet, shins, **nose, chin**, hands, knees and feet. Patches of hair loss were seen, limited to some of these depigmented areas over the vertex and occipital region of the scalp and eyebrows [Figure 3]. Other body areas were not affected by patchy hair loss. Clinically, the diagnosis of vitiligo with AA was made. *(..)* Additionally, the basal layer of the epidermis was almost devoid of pigment, [Figure 5] confirming the diagnosis of vitiligo over the same site. *(..)* Both AA and vitiligo are clubbed under the spectrum of autoimmune disorders. *(..)* Our case lends support to the hypothesis that AA and vitiligo share a common pathogenic pathway including autoimmune response against some common antigens like those derived from the bulb melanocytes. Melanocytes-derived peptide antigens released during vitiligo pathogenesis could act as auto-antigens not only for vitiligo, but also for AA, and autoimmune Th-cells against them could also trigger a response against the hair follicle melanocytes, thus, pre-disposing to AA. *(..text omitted..)*

Example 2 Original abstract: Both alopecia areata (AA) and vitiligo are autoimmune diseases, and their coexistence in the same patient is not uncommon, as vitiligo has been reported to occur in 4.1% of patients of AA. We present a case of a 15-year-old male child who had vitiligo and later developed AA over the existing lesions of vitiligo over face and scalp and have attempted to elucidate the current understanding of mechanisms of coexistence of these two diseases. Our case lends support to the hypothesis that AA and vitiligo share a common pathogenic pathway including autoimmune response against some common antigens like those derived from the bulb melanocytes. Stimulation of proinflammatory T-helper(Th)-1 cell mediated immunological response or inactivation of a suppressor T-cell mediated response could be the common underlying phenomenon. However, the striking rarity of colocalization of these two diseases has led to the recent debate over current understanding of their pathogenesis, and whether this association is merely a coincidence. As both AA and vitiligo are frequent and chronic dermatological disorders, it is of utmost importance to gain more understanding into their pathogenic mechanisms so that more definitive treatment modalities may be devised and the quality of life of these patients can be improved.

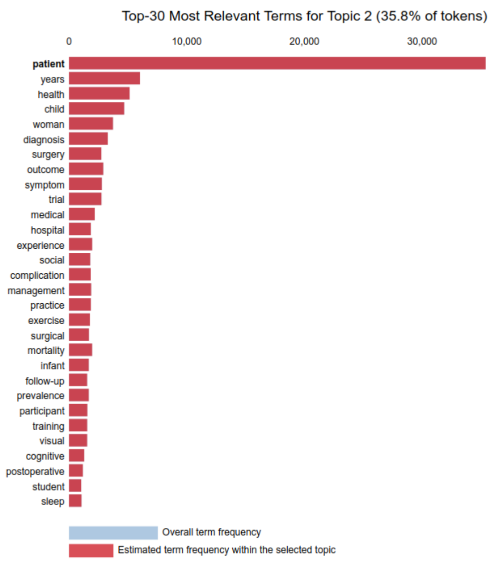
Example 2 fconv: alopecia areata (aa) and vitiligo are autoimmune diseases , and their coexistence in the same patient is not uncommon , as vitiligo has been reported to occur in 4.1 % of patients of aa and is about 4 times more common in patients with aa than in the general population . we present a case of a 15- year-old male child who had vitiligo and later developed aa over the scalp , forehead , eyebrows , eyebrows , perioral , preauricular regions , nose and chin depigmented macules over the scalp , forehead , eyebrows , eyebrows , perioral , preauricular regions , nose and chin depigmented macules over the scalp , forehead , eyebrows , periorbital , perioral , preauricular regions , nose and chin depigmented macules over the scalp , forehead , eyebrows , periorbital , perioral , preauricular regions , nose and chin depigmented macules over the scalp , forehead , eyebrows , periorbital



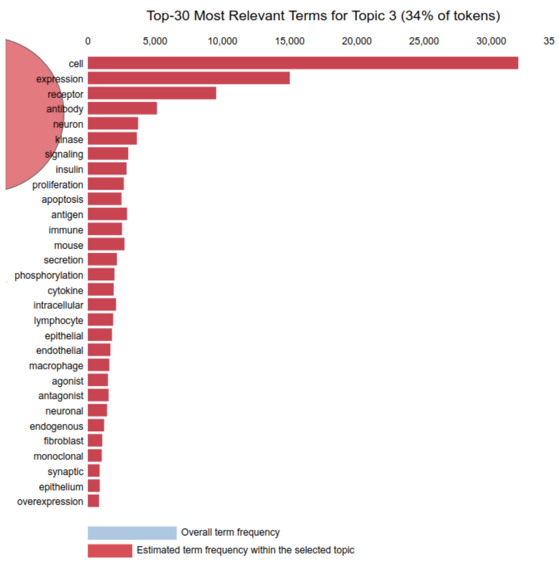
(a) Top-30 Most Relevant Terms in Overall



(b) Top-30 Most Relevant Terms for Topic 1



(c) Top-30 Most Relevant Terms for Topic 2



(d) Top-30 Most Relevant Terms for Topic 3

Figure 3.3: Top 30 most relevant terms for LDA topic modelling, computed on the abstracts from the *abstract-gen* dataset, using 3 LDA topics.

3.2 Summary Refinement through Denoising

Originally published as: Nikolov, N. I., Calmanovici, A., and Hahnloser, R. (2019). Summary refinement through denoising. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2019*

Abstract

We propose a simple method for post-processing the outputs of a text summarization system in order to refine its overall quality. Our approach is to train text-to-text rewriting models to correct information redundancy errors that may arise during summarization. We train on synthetically generated noisy summaries, testing three different types of noise that introduce out-of-context information within each summary. When applied on top of extractive and abstractive summarization baselines, our summary denoising models yield metric improvements while reducing redundancy.[9]

3.2.1 Introduction

Text summarization aims to produce a shorter, informative version of an input text. While extractive summarization only selects important sentences from the input, abstractive summarization generates content without explicitly re-using whole sentences (Nenkova et al., 2011). In recent years, a number of successful approaches have been proposed for both extractive (Nallapati et al., 2017; Narayan et al., 2018b) and abstractive (Chen and Bansal, 2018b; Gehrmann et al., 2018) summarization paradigms. Despite these successes, many state-of-the-art systems remain plagued by overly high output redundancy (See et al. (2017b); see Figure 3.6), which we set out to reduce.

In this paper, we propose a simple method (Figure 3.4, Section 3.2.3) for post-processing the outputs of a text summarization system in order to improve their overall quality. Our approach is to train dedicated text-to-text rewriting models to correct errors that may arise during summarization, focusing specifically on reducing information redundancy within each individual summary. To achieve this, we synthesize from clean summaries noisy summaries that contain diverse information redundancy errors, such as sentence repetition and out-of-context information (Section 3.2.3).

In our experiments (Section 3.2.5), we show that denoising yields metric improvements and reduces redundancy when applied on top of several extractive and abstractive baselines. The generality of our method makes it a useful post-processing step applicable to any summarization system, that

[9] Code available at `https://github.com/ninikolov/summary-denoising`.

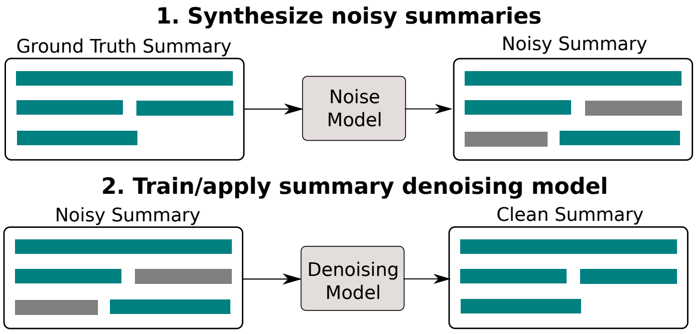


Figure 3.4: Overview of our approach to summary denoising. We alter ground truth summaries to generate a noisy dataset, on which we train denoising models to restore the original summaries.

standardizes the summaries and improves their overall quality, ensuring fewer redundancies across the text.

3.2.2 Background

Post-processing of noisy human or machine-generated text is a topic that has recently been gathering interest. Automatic error correction (Rozovskaya and Roth, 2016; Xie et al., 2018b) aims to improve the grammar or spelling of a text. In machine translation, automatic post editing of translated outputs (Chatterjee et al., 2018) is commonly used to further improve the translation quality, standardise the translations, or adapt them to a different domain (Isabelle, 2007).

In (Xie et al., 2018b), authors synthesize grammatically incorrect sentences from correct ones using backtranslation (Sennrich et al., 2016b), which they use for grammar error correction. They enforce hypothesis variety during decoding by adding noise to beam search. Another work that is close to ours is (Fevry and Phang, 2018), where authors introduce redundancy on the word level in order to build an unsupervised sentence compression system. In this work, we take a similar approach, but instead focus on generating information redundancy errors on the sentence rather than the word level.

3.2.3 Approach

Our approach to summary refinement consists of two steps. First, we use a dataset of clean ground truth summaries to **generate** noisy summaries using several different types of synthetic noise. Second, we train text rewriting models to correct and **denoise** the noisy summaries, restoring them to their original form. The learned denoising models are then used to post-process and refine the outputs of a summarization system.

Generating Noisy Summaries

To generate noisy datasets, we rely on an existing parallel dataset of articles and clean ground truth summaries $S = \{s_0, , ..., s_j\}$. We iterate over each of the summaries and perturb them with noise, according to a *sentence noise distribution* $p_{noise} = [p_0, p_1, ..., p_N]$. p_{noise} defines the probability of adding noise to a specific number of sentences within each summary (from 0 up to a maximum of N noisy sentences), with $\sum p_{noise} = 1$.

For all experiments in this work, we use $p_{noise} = [0.15, 0.85]$ in order to ensure consistency, meaning that ˜15% of our noisy summaries contain no noisy sentences, while ˜85% contain one noisy sentence. Initial experiments showed that distributions which enforce larger or smaller amounts of noise lead to stronger or weaker denoising effects. Our choice of noise distribution showed good results on the majority of systems that we tested; we leave a more rigorous investigation of the choice of distribution to future work.

In addition to adding noise, we generate 3 noisy summaries for each clean summary by picking multiple random sentences to noise. This step increases the dataset size while introducing variety.

Types of Noise

We experiment with three simple types of noise, all of which introduce *information redundancy* into a summary. Our aim is to train denoising models that minimize repetitive or peripheral information within summaries[10].

Repeat. picks random sentences from the summary and repeats them at the end. Repetition of phrases or even whole sentences is a problem commonly observed in text generation with RNNs (See et al., 2017b), which motivates efforts to detect and minimize repetitions.

Replace. picks random sentences from the summary, and replaces them with the closest sentence from the article. This type of noise helps the model to learn to *refine* sentences from the generated summaries, paraphrasing sentences when they are too long or contain redundant information.

[10] In preliminary experiments, we tested several other types of noise, such as word shuffling on the sentence or document level, or word deletion. Since we did not find them to be useful for reducing repetitions, or for increasing the quality of the summaries, we do not include these results here.

Extra. picks random sentences from the article, paraphrases them, and inserts them into the summary, preserving the order of the sentences as they appear in the article. With this type of noise, a model learns to delete sentences which are out of context or contain redundant information. To paraphrase the sentences, we use the sentence paraphrasing model from (Chen and Bansal, 2018b), trained on matching sentence pairs from the CNN/Daily Mail dataset.

Mixture. mixes all the above noise types uniformly into a single dataset, keeping the same dataset size as for the individual noise types. With mixture, we explore whether the benefits of each noise type can be combined into a single model.

3.2.4 Experimental Setup

Dataset. We use the CNN/Daily Mail dataset[11] (Hermann et al., 2015) of news articles and summaries in the form of bullet points, and follow the preprocessing pipeline from (Chen and Bansal, 2018b). We use the standard split of the dataset, consisting of 287k news-summary pairs for training and 13k pairs for validation. We follow Section 3.2.3 to generate noisy versions of the datasets to be used during training. During testing, instead of clean summaries that contain noisy sentences, we input summaries produced by existing extractive or abstractive summarization systems.

Denoising models. For all of our denoising experiments, we use a standard bidirectional LSTM encoder-decoder model (Sutskever et al., 2014) with 1000 hidden units and an attention mechanism (Bahdanau et al., 2015), and train on the subword-level (Sennrich et al., 2016c), capping the vocabulary size to 50k tokens for all experiments[12]. We train all models until convergence using the Adam optimizer (Kingma and Ba, 2015).

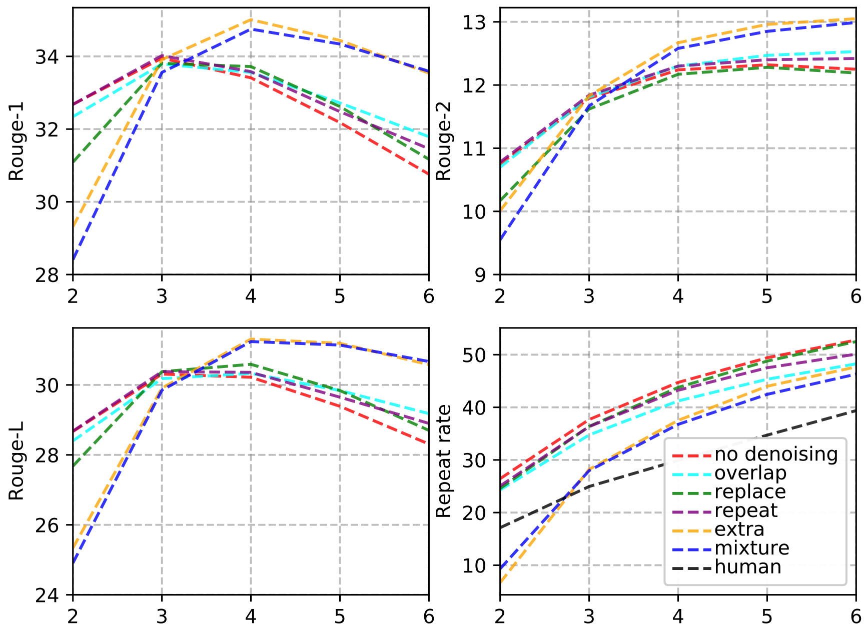
In addition to our neural denoising models, we implement a simple denoising baseline, `overlap`, based on unigram overlap between sentences in a summary. `overlap` deletes sentences which overlap more than 80%[13] with any other sentence in the summary and can therefore be considered as redundant.

Evaluation. We report the ROUGE-1/2/L metrics (Lin, 2004). We also report the *Repeat* rate (Nikolov et al., 2018a) $rep(\mathbf{s}) = \frac{\sum_i o(\overline{\mathbf{s}_i}, s_i)}{|\mathbf{s}|}$ which is the

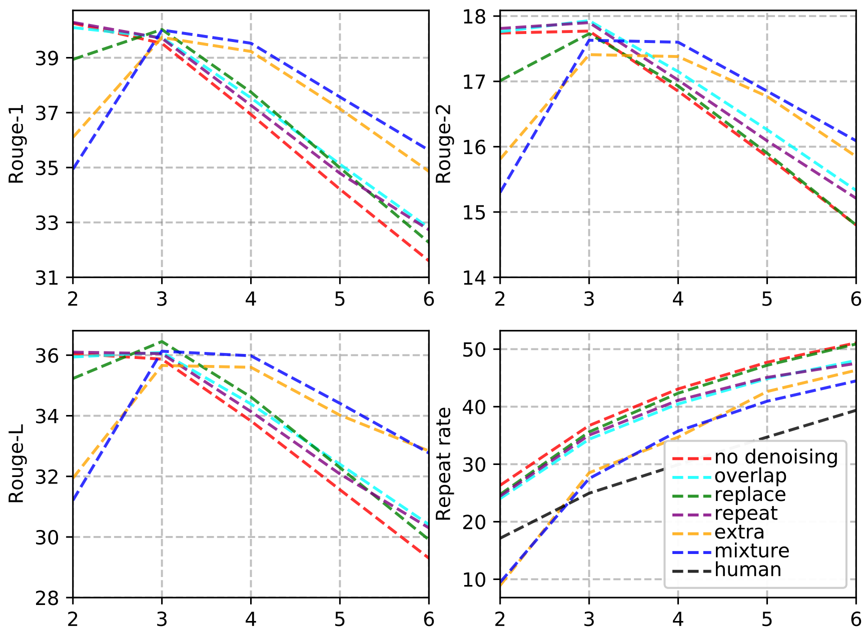
[11] `https://github.com/abisee/cnn-dailymail`

[12] We use the `fairseq` library `https://github.com/pytorch/fairseq`

[13] We empirically found that this threshold is sufficiently high to prevent unnecessary deletion and sufficiently low to detect near-identical sentences.



(a) Denoising the LexRank system.



(b) Denoising the RNN-Ext system.

Figure 3.5: Metric results (Rouge-1/2/L and Repeat rate) on denoising extractive summarization systems. The x-axis in all plots is the number of extracted sentences. `human` is the result of the ground truth summaries (only for the Repeat rate).

System	Den. approach	R-1	R-2	R-L	Repeat	#Sent	#Tok
Human	-	-	-	-	28.86	3.88	61.21
Article	-	14.95	8.54	14.41	70.5	26.9	804
Article	Mixture	30.47	13.97	28.24	53.43	10.67	304.7
RNN	-	35.61	15.04	32.7	51.9	2.93	58.46
RNN	Overlap	36.41	15.92	33.73	26.84	2.39	47.31
RNN	Repeat	**36.5**	**15.94**	**33.79**	27.65	2.41	48.34
RNN	Replace	35.2	14.86	32.4	51.51	2.98	57.0
RNN	Extra	33.95	14.58	31.2	37.19	2.21	42.82
RNN	Mixture	35.08	15.3	32.44	27.27	2.2	42.14
RNN-RL	-	**40.88**	**17.8**	**38.54**	39.29	4.93	72.82
RNN-RL	Overlap	40.76	17.69	38.43	37.71	4.83	71.02
RNN-RL	Repeat	40.84	17.76	38.49	38.78	4.86	71.69
RNN-RL	Replace	40.78	17.72	38.46	39.24	4.93	72.2
RNN-RL	Extra	39.12	16.7	36.76	34.04	3.84	55.43
RNN-RL	Mixture	40.11	17.33	37.76	35.45	4.18	61.15

Table 3.5: Results on denoising abstractive summarization. **R-1/2/L** are the ROUGE-1/2/L scores, **Repeat** is the Repeat rate, while **#Sent** and **#Tok** are the average numbers of sentences or tokens in the summaries. Best **ROUGE** results for each model are in bold. *Human* is the result of the ground truth summaries, while *Article* uses the original article as the summary.

average unigram overlap o of each sentence s_i in a text with the remainder of the text (where $\overline{\mathbf{s}_i}$ denotes the complement of sentence s_i). Since the repeat rate measures the overlapping information across all sentences in a summary, lower values signify that a summary contains many unique sentences, while higher values indicate potential information repetition or redundancy within a summary.

3.2.5 Results

Extractive Summarization

We experiment with denoising two extractive systems: LexRank (Erkan and Radev, 2004) is an unsupervised graph-based approach which measures the centrality of each sentence with respect to the other sentences in the document. RNN-Ext is a more recent supervised LSTM sentence extractor module from (Chen and Bansal, 2018b), trained on the CNN/Daily Mail dataset. It extracts sentences from the article sequentially. Both extractive systems require the number of sentences to be extracted to be given as a hyperparameter, in our experiments we test with summary lengths ranging from 2 to 6 sentences[14].

[14] The average sentence count of a summary in the CNN/Daily Mail dataset is 3.88.

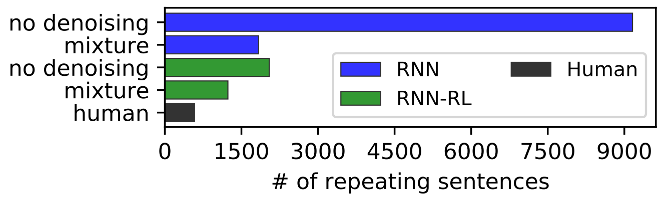


Figure 3.6: Number of sentence repetitions before and after denoising.

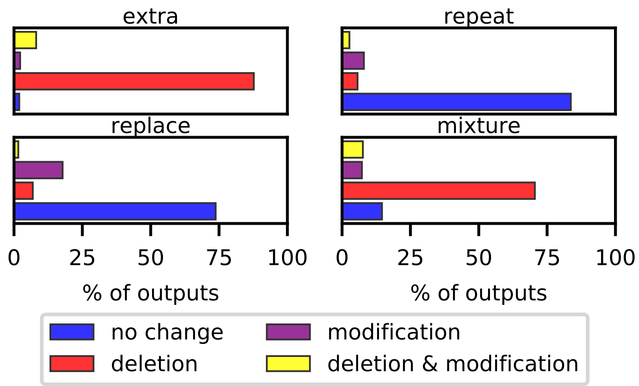
The results on extractive summarization are in Figure 3.5a for *LexRank* and Figure 3.5b for *RNN-Ext*, where we plot the metric scores for varying numbers of extracted sentences for each of the two systems. For both LexRank and RNN-Ext, we observe ROUGE improvements after denoising over the baseline systems without denoising. The `repeat` and `replace` methods yielded more modest improvements of 0.5-1 ROUGE-L points, performing comparably to the simple `overlap` baseline. The most effective noise types are `extra` and `mixture`, yielding improvements of up to 2 ROUGE-L points for LexRank and up to 3.5 ROUGE-L points for RNN-Ext. The superior performance to `overlap` indicates that the additional denoising operations learned by our models (see Figure 3.7a) are beneficial and can lead to more polished summaries that also may contain abstractive elements.

The gains from denoising are greater for longer summaries of more than two sentences. Long summaries are more likely to be affected by redundancy. For shorter summaries, denoising might lead to deletion of important information, thus denoising needs to be applied more carefully in such cases. Furthermore, for all sentence lengths and noise types, we observe a reduction in the Repeat rate after denoising, demonstrating that our approach is effective at reducing redundancy.

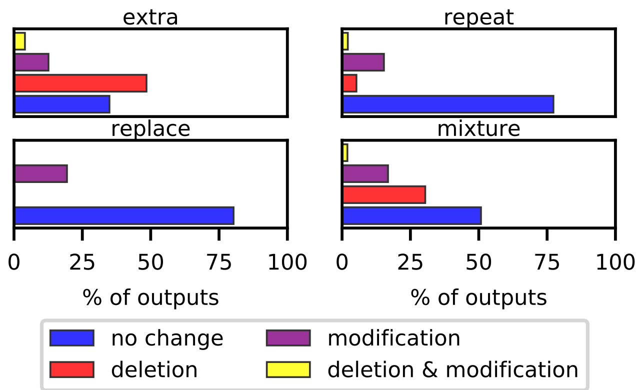
In Table 3.5, we additionally include the result from using the whole articles (*Article*) as input to our `mixture` model. Denoising is effective in this case, indicating that our approach may be promising for developing abstractive summarization systems that are fully unsupervised, similar to recent work in unsupervised sentence compression (Fevry and Phang, 2018).

Abstractive Summarization

For abstractive summarization, we test two systems. The first is a standard LSTM encoder-decoder model with an attention mechanism (*RNN*), identical to our denoising network from Section 3.2.4. The second, *RNN-RL*, is a state-of-the-art abstractive system proposed in (Chen and Bansal, 2018b) that combines extractive and abstractive summarization using reinforcement learning. We train *RNN* ourselves, while for *RNN-RL*, we use the outputs provided by the authors.



(a) RNN-Ext extractive system, extracting 5 sentences.



(b) RNN abstractive system.

Figure 3.7: Types of denoising operations applied to an extractive and an abstractive system (averaged over our test set).

Our metric results from denoising abstractive summarization are in Table 3.5. In Figure 3.6, we also compute the approximate number of sentence repetitions on the test set, by calculating the number of sentences that overlap significantly ($> 80\%$) with at least one other sentence in the summary.

For the *RNN* model, the `repeat` noise helps to remove repetition, halving our repetition metric, while boosting the ROUGE scores. This result is similar to our much simpler `overlap` baseline based on sentence deletion. The other noise types help to reduce redundancy, bringing the *Repeat* rate closer to that of *Human* summaries. This, however, comes at the cost of a decrease in ROUGE. For *RNN-RL*, while denoising helps to reduce repetition, none of our noise types managed to yield ROUGE improvements. One reason for this may be that this model already comes with a built-in mechanism for reducing redundancy which relies on sentence reranking (Chen and Bansal, 2018b). However, as shown in Figure 3.6 (and in our example in Table 3.6), this model still generates many more sentence repetitions than found in human summaries. In overall, our approach is effective at reducing redundant information in abstractive summaries, however this comes with a potential loss of information, which can lead to a reduction in ROUGE. Thus, our denoising methods are currently better suited for extractive than for absctractive summarization. Our work therefore calls for the development of novel types of synthetic noise that target abstractive summarization.

Analysis of Model Outputs

In Table 3.6 we show example outputs from denoising extractive and abstractive summaries produced for a sports article from our test set. All baseline summarization systems produced outputs that contain redundancy: for example, the first three sentences generated by the *RNN* system, and the 3rd and 4th sentences produced by the *RNN-RL* system are almost identical. To denoise the summaries, our models used diverse operations such as deletion of one or two sentences (e.g. *RNN* system, *Repeat* noise), rewriting (e.g. *RNN-RL* system, *Replace* noise, where *"dinorah santana , the player 's agent , said her client had rejected the offer of a three-year contract extension"* is paraphrased to *"the player 's agent said she had rejected the offer of a three-year contract"*), or even a combination of deletion and rewriting (e.g. *RNN-RL* system, *Repeat* noise).

3.2.6 Conclusion

We proposed a general framework for improving the outputs of a text summarization system based on denoising. Our approach is independent of the type of the system, and is applicable to both abstractive and extractive summarization paradigms. It could be useful as a post-processing step in a text summarization pipeline, ensuring that the summaries meet specific standards related to length or quality.

Our approach is effective at reducing information repetition present in existing summarization systems, and can even lead to ROUGE improvements, especially for extractive summarization. Denoising abstractive summarization proved to be more challenging, and our simple noise types did not yield significant ROUGE improvements for a state-of-the-art system. Our focus in future work, will, therefore, be to estimate better models of the noise present in abstractive summarization, to reduce information redundancy without a loss in quality, as well as to target other aspects such as the grammaticality or cohesion of the summary.

Table 3.6: Examples for denoising extractive and abstractive summarization. *Same* indicates a summary sentence has been unchanged, while *Deleted* indicates sentence deletion. In brackets, *R-1* denotes the *Rouge-1* score, while *Rep* denotes the *Repeat* rate.

Ground truth *(Rep=38.38)*: 1. dani alves has spent seven seasons with the catalan giants 2. alves has four spanish titles to his name with barcelona 3. the brazil defender has also won the champions league twice with barca		
RNN-Ext-4	**RNN**	**RNN-RL**
No denoising (R-1=33.6,Rep=45): 1. dani alves looks set to leave barcelona this summer after his representative confirmed the brazilian right-back had rejected the club 's final contract offer 2. alves has enjoyed seven successful years at barcelona , winning four spanish titles and the champions league twice 3. but the 31-year-old has been unable to agree a new deal with the catalan club and will leave the nou camp this summer 4. dinorah santana , the player 's agent and ex-wife , said at a press conference on thursday that her client had rejected the offer of a three-year contract extension , which was dependent on the player taking part in 60 per cent of matches for the club	*No denoising (R-1=34,Rep=79.6):* 1. dani alves has been unable to agree a new deal with catalan club 2. the brazilian has been unable to agree a new deal with catalan club 3. alves has been unable to agree a new deal with catalan club 4. alves has been linked with a number of clubs including manchester united and manchester city	*No denoising (R-1=31,Rep=51.6):* 1. dani alves looks set to leave barcelona this summer 2. alves has enjoyed seven successful years at barcelona 3. alves has been unable to agree a deal with the catalan club 4. the 31-year-old has been unable to agree a new deal 5. dinorah santana , the player 's agent , said her client had rejected the offer of a three-year contract extension
Replace (R-1=36.6,Rep=46.6): 1. Same 2. Same 3. Same 4. the player 's agent and ex-wife said at a press conference on thursday that her client had rejected the offer of a three-year contract extension	*Replace (R-1=34, Rep=79.6):* 1. Same 2. Same 3. Same 4. Same	*Replace (R-1=31, Rep=52.6):* 1. Same 2. Same 3. Same 4. Same 5. the player 's agent said she had rejected the offer of a three-year contract
Repeat (R-1=33.6,Rep=45): 1. Same 2. Same 3. Same 4. Same	*Repeat (R-1=28, Rep=41.4):* 1. Same 2. Deleted 3. Deleted 4. Same	*Repeat (R-1=24.2, Rep=36.1):* 1. Same 2. Same 3. Deleted 4. alves has been unable to agree a new deal 5. Same
Extra (R-1=43.6,Rep=36.8): 1. Same 2. Same 3. the 31-year-old has been unable to agree a new deal with the catalan club and will leave the nou camp this summer 4. Deleted	*Extra (R-1=37.2,Rep=92.8):* 1. Same 2. Same 3. Same 4. Deleted	*Extra (R-1=37, Rep=60.8):* 1. Same 2. Same 3. Same 4. Same 5. Deleted
Mixture (R-1=43, Rep=36.2): 1. Same 2. Same 3. Same 4. Deleted	*Mixture (R-1=28, Rep=41.43):* 1. Same 2. Deleted 3. Deleted 4. Same	*Mixture (R-1=37, Rep=60.8):* 1. Same 2. Same 3. Same 4. Same 5. Deleted

Chapter 4

Paraphrasing and Summarization without Parallel Data

4.1 Large-Scale Hierarchical Alignment for Data-driven Text Rewriting

Originally published as:

Abstract

We propose a simple unsupervised method for extracting pseudo-parallel monolingual sentence pairs from comparable corpora representative of two different text styles, such as news articles and scientific papers. Our approach does not require a seed parallel corpus, but instead relies solely on hierarchical search over pre-trained embeddings of documents and sentences. We demonstrate the effectiveness of our method through automatic and extrinsic evaluation on text simplification from the normal to the Simple Wikipedia. We show that pseudo-parallel sentences extracted with our method not only supplement existing parallel data, but can even lead to competitive performance on their own.[1]

4.1.1 Introduction

Parallel corpora are indispensable resources for advancing monolingual and multilingual text rewriting tasks. Due to the scarce availability of parallel corpora, and the cost of manual creation, a number of methods have been proposed that can perform large-scale sentence alignment: automatic extraction of *pseudo-parallel* sentence pairs from raw, comparable[2] corpora. While pseudo-parallel data is beneficial for machine translation (Munteanu and

[1] Code available at `https://github.com/ninikolov/lha`.
[2] Corpora that contain documents on similar topics.

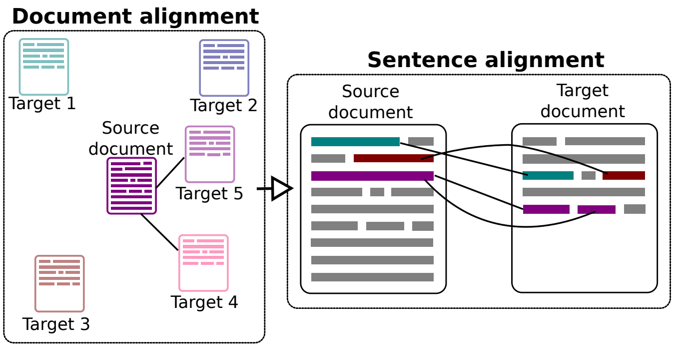


Figure 4.1: Illustration of large-scale hierarchical alignment (**LHA**). For each document in a *source* dataset, **document alignment** retrieves matching documents from a *target* dataset. In turn, **sentence alignment** retrieves matching sentence pairs from within each document pair.

Marcu, 2005), there has been little work on large-scale sentence alignment for monolingual text-to-text rewriting tasks, such as simplification (Nisioi et al., 2017) or style transfer (Liu et al., 2016). Furthermore, the majority of existing methods (e.g. (Marie and Fujita, 2017; Grégoire and Langlais, 2018)) assume access to some parallel training data. This impedes their application to cases where there is *no parallel data available whatsoever*, which is the case for the majority of text rewriting tasks, such as style transfer.

In this paper, we propose a simple unsupervised method, Large-scale Hierarchical Alignment (**LHA**) (Figure 4.1; Section 4.1.3), for extracting pseudo-parallel sentence pairs from two raw monolingual corpora which contain documents in two different author styles, such as scientific papers and press releases. LHA hierarchically searches for document and sentence nearest neighbors within the two corpora, extracting sentence pairs that have high semantic similarity, yet preserve the stylistic characteristics representative of their original datasets. LHA is robust to noise, fast and memory efficient, enabling its application to datasets on the order of hundreds of millions of sentences. Its generality makes it relevant to a wide range of monolingual text rewriting tasks.

We demonstrate the effectiveness of LHA on automatic benchmarks for alignment (Section 4.1.4), as well as extrinsically, by training neural machine translation (NMT) systems on the task of text simplification from the normal Wikipedia to the Simple Wikipedia (Section 4.1.5). We show that pseudo-parallel datasets obtained by LHA are not only useful for augmenting existing parallel data, boosting the performance on automatic measures, but can even be competitive on their own.

4.1.2 Background

Data-Driven Text Rewriting

The goal of text rewriting is to transform an input text to satisfy specific constraints, such as simplicity (Nisioi et al., 2017) or a more general author style, such as political (e.g. democratic to republican) or gender (e.g. male to female) (Prabhumoye et al., 2018; Shen et al., 2017). Rewriting systems can be valuable when preparing a text for multiple audiences, such as simplification for language learners (Siddharthan, 2002) or people with reading disabilities (Inui et al., 2003). They can also be used to improve the accessibility of technical documents, e.g. to simplify terms in clinical records for laymen (Abrahamsson et al., 2014).

Text rewriting can be cast as a data-driven task in which transformations are learned from large collections of parallel sentences. Limited availability of high-quality parallel data is a major bottleneck for this approach. Recent work on Wikipedia and the Simple Wikipedia (Coster and Kauchak, 2011; Kajiwara and Komachi, 2016) and on the Newsela dataset of simplified news articles for children (Xu et al., 2015b) explore supervised, data-driven approaches to text simplification. Such approaches typically rely on statistical (Xu et al., 2016) or neural (Stajner and Nisioi, 2018) machine translation.

Recent work on unsupervised approaches to text rewriting without parallel corpora is based on variational (Fu et al., 2017) or cross-aligned (Shen et al., 2017) autoencoders that learn latent representations of content separate from style. In (Prabhumoye et al., 2018), authors model style transfer as a back-translation task by translating input sentences into an intermediate language. They use the translations to train separate English decoders for each target style by combining the decoder loss with the loss of a style classifier, separately trained to distinguish between the target styles.

Large-Scale Sentence Alignment

The goal of **sentence alignment** is to extract from raw corpora sentence pairs suitable as training examples for text-to-text rewriting tasks such as machine translation or text simplification. When the documents in the corpora are *parallel* (labelled document pairs, such as identical articles in two languages), the task is to identify suitable sentence pairs from each document. This problem has been extensively studied both in the multilingual (Brown et al., 1991; Moore, 2002) and monolingual (Hwang et al., 2015; Kajiwara and Komachi, 2016; Stajner et al., 2018) case. The limited availability of parallel corpora led to the development of **large-scale sentence alignment** methods, which is also the focus of this work. The aim of these methods is to extract pseudo-parallel sentence pairs from raw, non-aligned corpora.

For many tasks, millions of examples occur naturally within existing textual resources, amply available on the internet.

The majority of previous work on large-scale sentence alignment is in machine translation, where adding pseudo-parallel pairs to an existing parallel dataset has been shown to boost the translation performance (Munteanu and Marcu, 2005; Uszkoreit et al., 2010). The work that is most closely related to ours is (Marie and Fujita, 2017), where authors use pre-trained word and sentence embeddings to extract rough translation pairs in two languages. Subsequently, they filter out low-quality translations using a classifier trained on parallel translation data. More recently, (Grégoire and Langlais, 2018) extract pseudo-parallel translation pairs using a Recurrent Neural Network (RNN) classifier. Importantly, these methods assume that *some parallel training data is already available*, which impedes their application in settings where there is no parallel data whatsoever, which is the case for many text rewriting tasks such as style transfer.

There is little work on large-scale sentence alignment focusing specifically on monolingual tasks. In (Barzilay and Elhadad, 2003), authors develop a hierarchical alignment approach of first clustering paragraphs on similar topics before performing alignment on the sentence level. They argue that, for monolingual data, pre-clustering of larger textual units is more robust to noise compared to fine-grained sentence matching applied directly on the dataset level.

4.1.3 Large-Scale Hierarchical Alignment (LHA)

Given two datasets that contain comparable documents written in two different author styles: a **source** dataset $\mathbf{S^d}$ consisting of N_S documents $\mathbf{S^d} = \{s_1^d, ..., s_{N_S}^d\}$ (e.g. all Wikipedia articles) and a **target** dataset $\mathbf{T^d}$ consisting of N_T documents $\mathbf{T^d} = \{t_1^d, ..., t_{N_T}^d\}$ (e.g. all articles from the Simple Wikipedia), our approach to large-scale alignment is hierarchical, consisting of two consecutive steps: **document** alignment followed by **sentence** alignment (see Figure 4.1).

Document Alignment

For each source document s_i^d, document alignment retrieves K nearest neighbours $\{t_{i_1}^d, ..., t_{i_K}^d\}$ from the target dataset. In combination, these form K pseudo-parallel document pairs $\{(s_i^d, t_{i_1}^d), ..., (s_i^d, t_{i_K}^d)\}$. Our aim is to select document pairs with high semantic similarity, potentially containing good pseudo-parallel sentence pairs representative of the document styles of each dataset.

To find nearest neighbours, we rely on two components: document embedding and approximate nearest neighbour search. For each dataset, we pre-compute document embeddings $e_d()$ as $I_s = [e_d(s_1^d), ..., e_d(s_{N_S}^d)]$ and $I_t = [e_d(t_1^d), ..., e_d(t_{N_T}^d)]$. We employ nearest neighbour search methods[3] to partition the embedding space, enabling fast and efficient nearest neighbour retrieval of similar documents across I_s and I_t. This enables us to find K nearest *target* document embeddings in I_t for each *source* embedding in I_s. We additionally filter document pairs whose similarity is below a manually selected threshold θ_d. In Section 4.1.4, we evaluate a range of different document embedding approaches, as well as alternative similarity metrics.

Sentence Alignment

Given a pseudo-parallel document pair (s^d, t^d) that contains a **source** document $s^d = \{s_1^s, ..., s_{N_J}^s\}$ consisting of N_J sentences and a **target** document $t^d = \{t_1^s, ..., t_{N_M}^s\}$ consisting of N_M sentences, sentence alignment extracts pseudo-parallel sentence pairs (s_i^s, t_j^s) that are highly similar.

To implement sentence alignment, we first embed each sentence in s^d and t^d and compute an inter-sentence similarity matrix P among all sentence pairs in s^d and t^d. From P we extract K nearest neighbours for each source and each target sentence. We denote the nearest neighbours of s_i^s as $NN(s_i^s) = \{t_{i_1}^s, \ldots, t_{i_K}^s\}$ and the nearest neighbours of t_j^s as $NN(t_j^s) = \{s_{j_1}^s, \ldots, s_{j_K}^s\}$. We remove all sentence pairs with similarity below a manually set threshold θ_s. We then merge *all* overlapping sets of nearest sentences in the documents to produce pseudo-parallel sentence sets (e.g. $(\{s_e^s, s_i^s\}, \{t_j^s, t_k^s, t_l^s\})$ when source sentence i is closes to target sentences j, k, and l and target sentence j is closest to source sentences e and i). This approach, inspired from (Stajner et al., 2018), provides the flexibility to model multi-sentence interactions, such as sentence splitting or compression, as well as individual sentence-to-sentence reformulations. Note that when $K = 1$, we only retrieve individual sentence pairs.

The final output of sentence alignment is a list of pseudo-parallel sentence pairs with high semantic similarity and preserved stylistic characteristics of each dataset. The pseudo-parallel pairs can be used to either augment an existing parallel dataset (as in Section 4.1.5), or independently, to solve a new author style transfer task for which there is no parallel data available (see the supplementary material for an example).

System Variants

The aforementioned framework provides the flexibility of exploring diverse variants, by exchanging document/sentence embeddings or text similarity metrics. We compare all variants in an automatic evaluation in Section 4.1.4.

[3] We use the Annoy library `https://github.com/spotify/annoy`.

Text embeddings We experiment with four text embedding methods:

1. *Avg*, is the average of the constituent word embeddings of a text[4], a simple approach that has proved to be a strong baseline for many text similarity tasks.
2. In *Sent2Vec*[5] (Pagliardini et al., 2018), the word embeddings are specifically optimized towards additive combinations over the sentence using an unsupervised objective function. This approach performs well on many unsupervised and supervised text similarity tasks, often outperforming more sophisticated supervised recurrent or convolutional architectures, while remaining very fast to compute.
3. *InferSent*[6] (Conneau et al., 2017) is a supervised sentence embedding approach based on bidirectional LSTMs, trained on natural language inference data.
4. *BERT*[7] (Devlin et al., 2019) is a state-of-the-art supervised sentence embedding approach based on the Transformer architecture.

Word Similarity We additionally test four word-based approaches for computing text similarity. Those can be used either on their own, or to refine the nearest neighbour search across documents or sentences.

1. We compute the unigram string overlap $o(\mathbf{x}, \mathbf{y}) = \frac{|\{\mathbf{y}\} \cap \{\mathbf{x}\}|}{|\{\mathbf{y}\}|}$ between source tokens $\mathbf{x}$ and target tokens $\mathbf{y}$ (excluding punctuation, numbers and stopwords).
2. We use the *BM25* ranking function (Robertson et al., 2009), an extension of TF-IDF.
3. We use the Word Mover's Distance (*WMD*) (Kusner et al., 2015), which measures the distance the embedded words of one document need to travel to reach the embedded words of another document. WMD has recently achieved good results on text retrieval (Kusner et al., 2015) and sentence alignment (Kajiwara and Komachi, 2016).
4. We use the Relaxed Word Mover's Distance (*RWMD*) (Kusner et al., 2015), which is a fast approximation of the WMD.

4.1.4 Automatic Evaluation

We perform an automatic evaluation of LHA using an annotated sentence alignment dataset (Hwang et al., 2015). The dataset contains 46 article pairs

[4] We use the Google News 300-dim Word2Vec models.
[5] We use the public unigram Wikipedia model.
[6] We use the GloVe-based model provided by the authors.
[7] We use the base 12-layer model provided by the authors.

Table 4.1: Automatic evaluation of Document (left) and Sentence alignment (right). **EDim** is the embedding dimensionality. **TP** is the percentage of true positives obtained at $\mathbf{F1}_{max}$. Speed is calculated on a single CPU thread.

			Document alignment				Sentence alignment			
	Approach	**EDim**	$\mathbf{F1}_{max}$	**TP**	θ_d	**doc/s**	$\mathbf{F1}_{max}$	**TP**	θ_s	**sent/s**
Embedding	Average word embeddings (Avg)	300	0.66	43%	0.69	260	0.675	46%	0.82	1458
	Sent2Vec Pagliardini et al. (2018)	600	**0.78**	**61%**	0.62	343	0.692	48%	0.69	1710
	InferSent† Conneau et al. (2017)	4096	-	-	-	-	0.69	49%	0.88	110
	BERT† Devlin et al. (2019)	768	-	-	-	-	0.65	43%	0.89	25
Word sim	Overlap	-	0.53	29%	0.66	120	0.63	40%	0.5	1600
	BM25 Robertson et al. (2009)	-	0.46	16%	0.257	60	0.52	27%	0.43	20K
	RWMD Kusner et al. (2015)	300	0.713	51%	0.67	60	0.704	50%	0.379	1050
	WMD Kusner et al. (2015)	300	0.49	24%	0.3	1.5	**0.726**	**54%**	0.353	180
	Hwang et al. (2015)	-	-	-	-	-	0.712	-	-	-
	Kajiwara and Komachi (2016)	-	-	-	-	-	0.724	-	-	-

†: These models are specifically designed for sentence embedding, hence we do not test them on document alignment.

Table 4.2: Evaluation on large-scale sentence alignment: identifying the *good* sentence pairs without any document-level information. We pre-compute the embeddings and use the Annoy ANN library. For the WMD-based approaches, we re-compute the top 50 sentence nearest neighbours of Sent2Vec.

Approach	$\mathbf{F1}_{max}$	TP	time
LHA (Sent2Vec)	0.54	31%	33s
LHA (Sent2Vec + WMD)	**0.57**	**33%**	1m45s
Global (Sent2Vec)	0.339	12%	15s
Global (WMD)	0.291	12%	30m45s

from Wikipedia and the Simple Wikipedia. The 67k potential sentence pairs were manually labelled as either *good* simplifications (277 pairs), *good* with a *partial* overlap (281 pairs), *partial* (117 pairs) or *non-valid*. We perform three comparisons using this dataset: evaluating document and sentence alignment separately, as well as jointly.

For sentence alignment, the task is to retrieve the 277 good sentence pairs out of the $67k$ possible sentence pairs in total, while minimizing the number of false positives. To evaluate document alignment, we add 1000 randomly sampled articles from Wikipedia and the Simple Wikipedia as noise, resulting in 1046 article pairs in total. The goal of document alignment is to identify the original 46 document pairs out of 1046×1046 possible document combinations.

This set-up additionally enables us to **jointly** evaluate document and sentence alignment, which best resembles the target effort of retrieving good sentence pairs from noisy documents. The two aims of the joint alignment task are to identify the *good* sentence pairs from within either $1M$ document or $125M$ sentence pairs, in the latter case without relying on any document-level information whatsoever.

Results

Our results are summarized in Tables 4.1 and 4.2. For all experiments, we set $K = 1$ and report the maximum F1 score ($\mathbf{F1}_{max}$) obtained from varying the document threshold θ_d and the sentence threshold θ_s. We also report the percentage of true positive (**TP**) document or sentence pairs that were retrieved when the F1 score was at its maximum, as well as the average speed of each approach (**doc/s** and **sent/s**). The speed becomes of a particular concern when working with large datasets consisting of millions of documents and hundreds of millions of sentences.

On document alignment, (Table 4.1, left) the *Sent2Vec* approach achieved the best score, outperforming the other embedding methods including the word-based similarity measures. On sentence alignment (Table 4.1, right), the WMD achieves the best performance, matching the result from (Kajiwara and Komachi, 2016). When evaluating document and sentence alignment jointly (Table 4.2), we compare our hierarchical approach (*LHA*) to global alignment applied directly on the sentence level (*Global*). *Global* computes the similarities between all 125M sentence pairs in the entire evaluation dataset. LHA significantly outperforms Global, successfully retrieving three times more valid sentence pairs, while remaining fast to compute. This result demonstrates that document alignment is beneficial, successfully filtering some of the noise, while also reducing the overall number of sentence similarities to be computed.

The *Sent2Vec* approach to LHA achieves good performance on document and sentence alignment, while also being the fastest to compute. We therefore use it as the default approach for the following experiments on text simplification.

4.1.5 Empirical Evaluation

To test the suitability of pseudo-parallel data extracted with LHA, we perform empirical experiments on text simplification from the normal Wikipedia to the Simple Wikipedia. We chose simplification because some parallel data are already available for this task, allowing us to experiment with mixing parallel and pseudo-parallel datasets. In the supplementary material we experiment with an additional task for which there is no parallel data: style transfer from scientific journal articles to press releases.

We compare the performance of neural machine translation (NMT) systems trained under three different scenarios: 1) using **existing** parallel data for training; 2) using a **mixture** of parallel and pseudo-parallel data extracted with LHA; and 3) using pseudo-parallel data **on its own**.

Experimental Setup

NMT model For all experiments, we use a single-layer LSTM encoder-decoder model (Cho et al., 2015) with an attention mechanism (Bahdanau et al., 2015). We train our models on the subword level (Sennrich et al., 2016c), capping the vocabulary size to 50k. We re-learn the subword rules separately for each dataset, and train until convergence using the Adam optimizer (Kingma and Ba, 2015). We use beam search with a beam of 5 to generate all final outputs.

Evaluation metrics We report a diverse range of automatic metrics and statistics. *SARI* (Xu et al., 2016) is a recently proposed metric for text simplification which correlates well with simplicity in the output. SARI takes into

Table 4.3: Datasets used to extract pseudo-parallel monolingual sentence pairs in our experiments.

Dataset	Type	Doc.	Tok.	Sent.	Tok. per sent.	Sent. per doc.
Wikipedia	Articles	5.5M	2.2B	92M	25 ± 16	17 ± 32
Simple Wikipedia	Articles	134K	62M	2.9M	27 ± 68	22 ± 34
Gigaword	News	8.6M	2.5B	91M	28 ± 12	11 ± 7

Table 4.4: Example pseudo-parallel pairs extracted by our Large-scale hierarchical alignment (LHA) method.

Dataset	Source	Target
wiki-simp-65	However, Jimmy Wales, Wikipedia's co-founder, denied that this was a crisis or that Wikipedia was running out of admins, saying, "The number of admins has been stable for about two years, there's really nothing going on."	But the co-founder Wikipedia, Jimmy Wales, did not believe that this was a crisis. He also did not believe Wikipedia was running out of admins.
wiki-news-74	Prior to World War II, Japan's industrialized economy was dominated by four major zaibatsu: Mitsubishi, Sumitomo, Yasuda and Mitsui.	Until Japan 's defeat in World War II , the economy was dominated by four conglomerates , known as " zaibatsu " in Japanese . These were the Mitsui , Mitsubishi , Sumitomo and Yasuda groups .

Table 4.5: Statistics of the pseudo-parallel datasets extracted with LHA. μ^{src}_{tok} and μ^{tgt}_{tok} are the mean src/tgt token counts, while $\%^{src}_{s>2}$ and $\%^{tgt}_{s>2}$ report the percentage of items that contain more than one sentence.

Dataset	Pairs	μ^{src}_{tok}	μ^{tgt}_{tok}	$\%^{src}_{s>2}$	$\%^{tgt}_{s>2}$
wiki-simp-72	25K	26.72	22.83	16%	11%
wiki-simp-65	80K	23.37	15.41	17%	7%
wiki-news-74	133K	25.66	17.25	19%	2%
wiki-news-70	216K	26.62	16.29	19%	2%

Table 4.6: Empirical results on text simplification from Wikipedia to the Simple Wikipedia. The highest SARI/BLEU results from each category are in bold. `input` and `reference` are not generated using Beam Search.

Method or Dataset	Total pairs (% pseudo)	Beam hypothesis 1					Beam hypothesis 2				
		SARI	BLEU	μ_{tok}	LD_{src}	LD_{tgt}	SARI	BLEU	μ_{tok}	LD_{src}	LD_{tgt}
input	-	26	99.37	22.7	0	0.26	-	-	-	-	-
reference	-	38.1	70.21	22.3	0.26	0	-	-	-	-	-
NTS	282K (0%)	30.54	84.69	-	-	-	35.78	77.57	-	-	-
Parallel + Pseudo-parallel or Randomly sampled data (Using full parallel dataset, 282K parallel pairs)											
baseline-282K	282K (0%)	30.72	85.71	18.3	0.18	0.37	36.16	82.64	19	0.19	0.36
+ wiki-simp-72	307K (8%)	30.2	87.12	19.43	0.14	0.34	36.02	81.13	19.03	0.19	0.36
+ wiki-simp-65	362K (22%)	**30.92**	**89.64**	19.8	0.13	0.33	36.48	83.56	19.37	0.18	0.35
+ wiki-news-74	414K (32%)	30.84	89.59	19.67	0.13	0.33	36.57	**83.85**	19.13	0.18	0.35
+ wiki-news-70	498K(43%)	30.82	89.62	19.6	0.13	0.33	36.45	83.11	18.98	0.19	0.36
+ rand-100K	382K (26%)	30.52	88.46	19.7	0.14	0.34	**36.96**	82.86	19	0.2	0.36
+ rand-200K	482K (41%)	29.47	80.65	19.3	0.18	0.36	34.36	74.67	18.93	0.23	0.38
+ rand-300K	582K (52%)	28.68	75.61	19.57	0.23	0.4	32.34	68.9	18.35	0.3	0.43
Parallel + Pseudo-parallel data (Using partial parallel dataset, 71K parallel pairs)											
baseline-71K	71K (0%)	**31.16**	69.53	17.45	0.29	0.44	32.92	67.29	19.14	0.3	0.44
+ wiki-simp-65	150K (52%)	31.0	**81.52**	18.26	0.21	0.38	**35.12**	**77.38**	18.16	0.25	0.39
+ wiki-news-70	286K(75%)	31.01	80.03	17.82	0.23	0.4	34.14	76.44	17.31	0.28	0.43
Pseudo-parallel data only											
wiki-simp-all	104K (100%)	29.93	60.81	18.05	0.36	0.47	30.13	57.46	18.53	0.39	0.49
wiki-news-all	348K (100%)	22.06	28.51	13.68	0.6	0.63	23.08	29.62	14.01	0.6	0.64
pseudo-all	452K (100%)	**30.24**	**71.32**	17.82	0.3	0.43	**31.41**	**65.65**	17.65	0.33	0.45

account the total number of changes (additions, deletions) of the input when scoring model outputs. *BLEU* (Papineni et al., 2002) is a precision-based metric for machine translation commonly used for evaluation of text simplification (Xu et al., 2016; Stajner and Nisioi, 2018) and of style transfer (Shen et al., 2017). Recent work has indicated that BLEU is not suitable for assessment of simplicity (Sulem et al., 2018), it correlates better with meaning preservation and grammaticality, in particular when using multiple references. We also report the average Levenshtein distance (LD) from the model outputs to the input (LD_{src}) or the target reference (LD_{tgt}). On simplification tasks, LD correlates well with meaning preservation and grammaticality (Sulem et al., 2018), complementing BLEU.

Extracting pseudo-parallel data We use LHA with *Sent2Vec* (see Section 4.1.3) to extract pseudo-parallel sentence pairs for text simplification. To ensure some degree of lexical similarity, we exclude pairs whose string overlap (defined in Section 4.1.3) is below 0.4, and pairs in which the target sentence is more than 1.5 times longer than the source sentence. We use $K = 5$ in all of our alignment experiments, which enables extraction of up to 5 sentence nearest neighbours.

Parallel data As a parallel baseline dataset, we use an existing dataset from (Hwang et al., 2015). The dataset consists of 282K sentence pairs obtained after aligning the parallel articles from Wikipedia and the Simple Wikipedia. This dataset allows us to compare our results to previous work on data-driven text simplification. We use two versions of the dataset in our experiments: `full` contains all 282K pairs, while `partial` contains 71K pairs, or 25% of the full dataset.

Evaluation data We evaluate our simplification models on the testing dataset from (Xu et al., 2016), which consists of 358 sentence pairs from the normal Wikipedia and the Simple Wikipedia. In addition to the ground truth simplifications, each input sentence comes with 8 additional references, manually simplified by Amazon Meachanical Turkers. We compute *BLEU* and *SARI* on the 8 manual references.

Pseudo-parallel data We align two dataset pairs, obtaining pseudo-parallel sentence pairs for text simplification (statistics of the datasets we use for alignment are in Table 4.10). First, we align the normal `Wikipedia` to the `Simple Wikipedia` using document and sentence similarity thresholds $\theta_d = 0.5$ and $\theta_s = \{0.72, 0.65\}$, producing two datasets: `wiki-simp-72` and `wiki-simp-65`. Because LHA uses no document-level information in this dataset, alignment leads to new sentence pairs, some of which may be distinct from the pairs present in the existing parallel dataset. We monitor for and exclude pairs that overlap with the testing dataset. Second, we align `Wikipedia` to the `Gigaword` news article corpus (Napoles et al., 2012), using $\theta_d = 0.5$ and $\theta_s = \{0.74, 0.7\}$, resulting in two additional pseudo-parallel datasets: `wiki-news-74` and `wiki-news-70`. With these datasets, we investigate whether pseudo-parallel data extracted from a *different domain* can be beneficial for text simplification. We use slightly higher sentence alignment thresholds for the news articles because of the domain difference.

We find that the majority of the pairs extracted contain a single sentence, and 15-20% of the source examples and 5-10% of the target examples contain multiple sentences (see Table 4.9 for additional statistics). Most multi-sentence examples contain two sentences, while 0.5-1% contain 3 to 5 sentences. Two example aligned outputs are in Table 4.4 (additional examples are available in the supplementary material). They suggest that our method is capable of extracting high-quality pairs that are similar in meaning, even spanning across multiple sentences.

Randomly sampled pairs We also experiment with adding random sentence pairs to the parallel dataset (`rand-100K`, `rand-200K` and `rand-300K` datasets, containing 100K, 200K and 300K random pairs, respectively). The random pairs are uniformly sampled from the Wikipedia and the Simple Wikipedia, respectively. With the random pairs, we aim to investigate how model performance changes as we add an increasing number of sentence pairs that are non-parallel but are still representative of the two dataset styles.

Automatic Evaluation

The simplification results in Table 4.6 are organized in several sections according to the type of dataset used for training. We report the results of the top two beam search hypotheses produced by our models, considering that the second hypothesis often generates simpler outputs (Stajner and Nisioi, 2018).

In Table 4.6, `input` is copying the normal Wikipedia input sentences, without making any changes. `reference` reports the score of the original Simple Wikipedia references with respect to the other 8 references available for this dataset. `NTS` is the previously best reported result on text simplification using neural sequence models (Stajner and Nisioi, 2018). `baseline-{282K, 71K}` are our parallel LSTM baselines, trained on 282K and 71K parallel pairs, respectively.

The models trained on a *mixture* of *parallel* and *pseudo-parallel* data generate longer outputs on average, and their output is more similar to the `input`, as well as to the original Simple Wikipedia `reference`, in terms of the LD. Adding pseudo-parallel data frequently yields BLEU improvements on both

Beam hypotheses: over the *NTS* system, as well as over our baselines trained solely on parallel data. The BLEU gains are larger when using the smaller parallel dataset, consisting of 71K sentence pairs. In terms of SARI, the scores remain either similar or slightly better than the baselines, indicating that simplicity in the output is preserved. The second Beam hypothesis yields higher SARI scores than the first one, in agreement with (Stajner and Nisioi, 2018). Interestingly, adding out-of-domain pseudo-parallel news data (`wiki-news-*` datasets) results in an increase in BLEU despite the potential change in style of the target sequence.

Larger pseudo-parallel datasets can lead to bigger improvements, however noisy data can result in a decrease in performance, motivating careful data selection. In our *parallel and random* set-up, we find that an increasing number of random pairs added to the parallel data progressively degrades model performance. However, those models still manage to perform surprisingly well, even when over half of the pairs in the dataset are random. Thus, neural machine translation can successfully learn target transformations despite substantial data corruption, demonstrating robustness to noisy or non-parallel data for certain tasks.

When training solely on *pseudo-parallel* data, we observe lower performance on average in comparison to parallel models. However, the results are encouraging, demonstrating the potential of our approach in tasks for which there is no parallel data available. As expected, the out-of-domain news data (`wiki-news-all`) is less suitable for simplification than the in-domain data (`wiki-simp-all`), because of the change in output style of the former. Results are best when mixing all pseudo-parallel pairs into a single dataset (`pseudo-all`). Having access to a small amount of *in-domain* pseudo-parallel data, in addition to *out-of-domain* pairs, seems to be beneficial to the success of our approach.

Human Evaluation

Due to the challenges of automatic evaluation of text simplification systems (Sulem et al., 2018), we also perform a human evaluation. We asked 8 fluent English speakers to rate the grammaticality, meaning preservation, and simplicity of model outputs produced for 100 randomly selected sentences from our test set. We exclude any model outputs which leave the input unchanged. Grammaticality and meaning preservation are rated on a Likert scale from 1 (Very bad) to 5 (Very good). Simplicity of the output sentences, in comparison to the input, is rated following (Stajner et al., 2018), between: -2 (much more difficult), -1 (somewhat more difficult), 0 (equally difficult), 1 (somewhat simpler) and 2 (much simpler).

The results are reported in Table 4.7, where we compare our parallel baseline (`baseline-272K` in Table 4.6) to our best model trained on a mixture of par-

Table 4.7: Human evaluation of the Grammaticality (**G**), Meaning preservation (**M**) and Simplicity (**S**) of model outputs (on the first Beam hypothesis).

Method	G	M	S
reference	4.53	4.34	0.69
baseline-272K	4.51	3.68	0.9
+ wiki-simp-65	4.39	3.76	0.74
pseudo-all	4.02	2.96	0.77

Table 4.8: Example model outputs (first Beam hypothesis).

Method	Example
input	jeddah is the **principal** gateway to mecca , islam ' s holiest city , which able-bodied muslims are required to visit at least once in their lifetime .
reference	jeddah is the **main** gateway to mecca , the holiest city of islam , where able-bodied muslims must go to at least once in a lifetime .
baseline-282K	it is the highest gateway to mecca , islam .
+ wiki-sim-65	jeddah is the **main** gateway to mecca , islam 's holiest city .
+ wiki-news-74	it is the **main** gateway to mecca , islam ' s holiest city .
pseudo-all	islam is the **main** gateway to mecca , islam 's holiest city .

allel and pseudo-parallel data (`wiki-simp-65`) and our best model trained on pseudo-parallel data only (`pseudo-all`). We also evaluate the original Simple Wikipedia references (`reference`) for comparison. In terms of simplicity, our pseudo-parallel systems are closer to the result of `reference` than is `baseline-272K`, indicating that they better match the target sentence style. `baseline-272K` and `wiki-simp-65` perform similarly to the references in terms of grammaticality, with `baseline-272K` having a small edge. In terms of meaning preservation, both do worse than the references, with `wiki-simp-65` having a small edge. `pseudo-all` performs worse on both grammaticality and meaning preservation, but is on par with the simplicity result of `wiki-simp-65`.

In Table 4.8, we also show example outputs of our best models (additional examples are available in the supplementary material). The models trained on parallel plus additional pseudo-parallel data produced outputs that preserve the meaning of 'Jeddah' as a city better than our parallel baseline, while correctly simplifying *principal* to *main*. The model trained solely on pseudo-parallel data produces a similar output, apart from wrongly replacing *jeddah* with *islam*.

4.1.6 Conclusion

We developed a hierarchical method for extracting pseudo-parallel sentence pairs from two monolingual comparable corpora composed of different text styles. We evaluated the performance of our method on automatic alignment benchmarks and extrinsically on automatic text simplification. We find improvements arising from adding pseudo-parallel sentence pairs to existing parallel datasets, as well as promising results when using the pseudo-parallel data on its own.

Our results demonstrate that careful engineering of pseudo-parallel datasets can be a successful approach for improving existing monolingual text-to-text rewriting tasks, as well as for tackling novel tasks. The pseudo-parallel data could also be a useful resource for dataset inspection and analysis. Future work could focus on improvements of our system, such as refined approaches to sentence pairing.

4.1.7 Appendix A: Style transfer from papers to press releases

We additionally test our large-scale hierarhical alignment (LHA) method on the task of style transfer from scientific journal articles to press releases. This task is novel: there is currently no parallel data available for it. The task is well-suited for our system, as there are many open access scientific repositories with millions of articles freely available.

Evaluation dataset We download press releases from the EurekAlert[8] aggregator and identify the digital object identifyer (DOI) in the text of each press release using regular expressions. Given the DOIs, we query the full text of a paper using the Elsevier ScienceDirect API[9]. Using this approach, we were able to compose 26k parallel pairs of papers and their press releases. We then applied our sentence aligner with *Sent2Vec* to extract 11k parallel sentence pairs, which we use in the evaluation below.

Pseudo-parallel data We used LHA with *Sent2Vec* and nearest neighbour limit $K = 5$ to extract pseudo-parallel sentence pairs. We align $350k$ additional press releases from `EurekAlert`, for which we were unable to retrieve the full text of a paper, to two large repositories of scientific papers: `PubMed`[10], which contains the full text of $1.5M$ open access papers, and `Medline`[11], which contains over $17M$ scientific abstracts (see Table 4.10 for an overview of these datasets). After alignment using a document similarity threshold $\theta_d = 0.6$ and a sentence similarity threshold $\theta_s = 0.74$, we extracted

[8] `www.eurekalert.org`

[9] `https://dev.elsevier.com/sd_apis.html`

[10] `www.ncbi.nlm.nih.gov/pmc/tools/openftlist`

[11] `www.nlm.nih.gov/bsd/medline.html`

$80k$ pseudo-parallel pairs in total (`paper-press-74` dataset). We additionally aligned `PubMed` and `Medline` to all `Wikipedia` articles using $\theta_d = 0.6$ and $\theta_s = 0.78$, obtaining out-of-domain pairs for this task (`paper-wiki-78` dataset). Table 4.9 contains some statistics of these pseudo-parallel datasets.

Table 4.9: Statistics of the pseudo-parallel datasets extracted with LHA for the style transfer task. μ_{tok}^{src} and μ_{tok}^{tgt} are the mean src/tgt token counts, while $\%_{s>2}^{src}$ and $\%_{s>2}^{tgt}$ report the percentage of items that contain more than one sentence.

Dataset	**Pairs**	μ_{tok}^{src}	μ_{tok}^{tgt}	$\%_{s>2}^{src}$	$\%_{s>2}^{tgt}$
paper-press-74	80K	28.01	16.06	22%	1%
paper-wiki-78	286K	25.84	24.69	20%	11%

Automatic evaluation

Our results on this task are summarized in Table 4.11, where `input` is the score obtained from copying the input sentences, and `reference` is the score of the original press release references. In addition to the previously described automatic measures, we also report the classification prediction of a Convolutional Neural Network (CNN) sentence classifier, trained to distinguish between the two target styles in this task: papers vs. press releases. To obtain training data for the classifier, we randomly sample 1 million sentences each from the `PubMed` and `EurekAlert` datasets. The CNN model

Table 4.10: Datasets used to extract pseudo-parallel monolingual sentence pairs in our style transfer experiments.

Dataset	**Type**	**Doc.**	**Tok.**	**Sent.**	**Tok. p. s.**	**Sent. p. d.**
PubMed	Sci. papers	1.5M	5.5B	230M	24 ± 14	180 ± 98
MEDLINE	Sci. abstracts	16.8M	2.6B	118M	26 ± 13	7 ± 4
EurekAlert	Press releases	358K	206M	8M	29 ± 15	23 ± 13

Table 4.11: Empirical results on style transfer from scientific papers to press releases. **Classifier** denotes the percentage of model outputs that were classified to belong to the press release class. μ_{tok} is the mean token length of the model outputs, while LD is the Levenshtein distance of a model output, to the input (LD_{src}) or the expected output (LD_{tgt}) sentences.

Method	**Total pairs**	**SARI**	**BLEU**	**Classifier**	μ_{tok}	LD_{src}	LD_{tgt}
input	-	19.95	45.88	43.71%	31.54	0	0.39
reference	-	-	-	78.72%	26.0	0.39	0
Pseudo-parallel data only							
paper-press-74	80K	32.83	24.98	**82.01%**	19.43	0.49	0.56
paper-wiki-78	286K	33.67	28.87	69.1%	27.06	0.43	0.53
combined	366K	**35.27**	**34.98**	70.18%	22.98	0.37	0.49

achieves 94% classification accuracy on a held-out set. With the classifier, we aim to investigate to what extent the overall style of the press releases is captured by the model outputs.

All of our models outperform the `input` in terms of SARI and produce outputs that are closer to the target style, according to the classifier. The in-domain `paper-press` dataset performs worse than the out-of-domain `paper-wiki` dataset, most likely because of the larger size of the latter. `paper-wiki` generates longer outputs on average than `reference`, which may be a source of lower classification score. The best performance is achieved by the `combined` dataset, which also produces outputs that are the closest to `input` and `reference`, in terms of the Levenshtein distance (LD).

4.1.8 Appendix B: Additional examples of aligned sentences and model outputs

Table 4.12 contains additional example pseudo-parallel sentences, that were extracted with LHA, for both the text simplification and the style transfer task. Table 4.13 contains additional model outputs for text simplification, while Table 4.14 contains model outputs for style transfer from papers to press releases.

Table 4.12: Example pseudo-parallel pairs extracted by LHA.

Dataset	Source	Target
wiki-simp-65	The dish is often served shredded with a dressing of oil, soy sauce, vinegar and sugar, or as a salad with vegetables.	They are often eaten in a kind of salad, with soy sauce or vinegar.
wiki-simp-65	The standard does not define bit rates for transmission, except that it says it is intended for bit rates lower than 20,000 bits per second.	These speeds are raw bit rates (in Million bits per second).
wiki-simp-65	However, Jimmy Wales, Wikipedia's co-founder, denied that this was a crisis or that Wikipedia was running out of admins, saying, "The number of admins has been stable for about two years, there's really nothing going on."	But the co-founder Wikipedia, Jimmy Wales, did not believe that this was a crisis. He also did not believe Wikipedia was running out of admins.
wiki-news-74	Thailand's coup leaders Thursday banned political parties from holding meetings or from conducting any other activities, according to a statement read on national television. "Political gatherings of more than five people have already been banned, but political activities can resume when normalcy is restored," the statement said.	" In order to maintain law and order , meetings of political parties and conducting of other political activities are banned , " the council said in its televised statement . " Political activities can resume when normalcy is restored , " it said .
paper-press-74	Such studies have suggested that some pterosaurs may have fed like modern-day skimmers, a rarified group of shorebirds, belonging to the genera Rynchops, that fly along the surface of still bodies of water scooping up small fish and crustaceans with their submerged lower jaw.	Previous studies have suggested that some pterosaurs may have fed like modern-day 'skimmers', a rare group of shorebirds, belonging to the Rynchops group. These sea-birds fly along the surface of lakes and estuaries scooping up small fish and crustaceans with their submerged lower jaw.
paper-press-74	Obsessions are defined as recurrent, persistent, and intrusive thoughts, images, or urges that cause marked anxiety, and compulsions are defined as repetitive behaviors or mental acts that the patient feels compelled to perform to reduce the obsession-related anxiety [26].	Obsessions are recurrent and persistent thoughts, impulses, or images that are unwanted and cause marked anxiety or distress.
paper-wiki-78	Then height and weight were used to calculate body mass index (kg/m2) as a general measure of weight status.	Body mass index is a mathematical combination of height and weight that is an indicator of nutritional status.
paper-wiki-78	Men were far more likely to be overweight (21.7%, 21.3-22.1%) or obese (22.4%, 22.0-22.9%) than women (9.5% and 9.9%, respectively), while women were more likely to be underweight (21.3%, 20.9-21.7%) than men (5.9%, 5.6-6.1%).	A 2008 report stated that 28.6% of men and 20.6% of women in Japan were considered to be obese. Men were more likely to be overweight (67.7%) and obese (25.5%) than women (30.9% and 23.4% respectively).

Table 4.13: Examples for text simplification from Wikipedia to the Simple Wikipedia. **-H1* and **-H2* denote the first and second Beam outputs.

Method	**Example**
input	jeddah is the principal gateway to mecca , islam ' s holiest city , which able-bodied muslims are required to visit at least once in their lifetime .
reference	jeddah is the main gateway to mecca , the holiest city of islam , where able-bodied muslims must go to at least once in a lifetime .
our-baseline-full-H1	it is the highest gateway to mecca , islam .
our-baseline-full-H2	it is the main gateway to mecca .
wiki-sim-65-H1	jeddah is the main gateway to mecca , islam 's holiest city .
wiki-simp-65-H2	it is the main gateway to mecca , islam 's holiest city .
wiki-news-74-H1	it is the main gateway to mecca , islam ' s holiest city .
wiki-news-74-H2	the main gateway to mecca is islam ' s holiest city .
pseudo-all-H1	islam is the main gateway to mecca , islam 's holiest city .
pseudo-all-H2	islam is the holiest city in the world .
input	this closely resembles the unix philosophy of having multiple programs each doing one thing well and working together over universal interfaces .
reference	this is very similar to the unix philosophy of having multiple programs doing one thing well and working together over universal interfaces .
our-baseline-full-H1	this closely resembles the unix philosophy of having multiple programs each doing one thing well and working together over universal interfaces .
our-baseline-full-H2	this closely resembles the unix method of having multiple programs each doing one thing well and working together over universal interfaces .
wiki-simp-65-H1	this closely resembles the unix philosophy of having multiple programs each doing one thing well and working together over universal interfaces .
wiki-simp-65-H2	this closely resembles the unix philosophy of having multiple programs each doing one thing well .
wiki-news-74-H1	this closely resembles the unix philosophy of having multiple programs each doing one thing well and working together over universal interfaces .
wiki-news-74-H2	this closely resembles the unix philosophy of having multiple programs each doing one thing well .
pseudo-all-H1	this is like the unix philosophy of having multiple programs each doing one thing well and working together over universal interfaces .
pseudo-all-H2	this is like the unix philosophy of having multiple programs each doing one thing well .

Table 4.14: Examples for style transfer from papers to press releases.

Method	Example
input	here we report evidence for this hypothesis and show that age-related declines in rppc grey matter volume better account for age-related changes in risk preferences than does age per se .
reference	″ this means that gray matter volume accounts for age-related changes in risk attitude more than age itself ″ .
paper-press-74	the authors suggest that age-related declines in rfc grey matter volume better account for age-related changes in risk preferences than does age per se .
paper-wiki-78	″ we report evidence for this hypothesis and show that age-related declines in rypgrey matter volume better account for age-related changes in risk preferences than does age per se ″ .
combined	″ we report evidence that age-related declines in the brain volume better account for age-related changes in risk preferences than does age per se ″ .
input	it is one of the most common brain disorders worldwide , affecting approximately 15 % 20 % of the adult population in developed countries (global burden of disease study 2013 collaborators , 2015) .
reference	migraine is one of the most common brain disorders worldwide , affecting approximately 15-20 % of the adults in developed countries .
paper-press-74	it is one of the most common brain disorders in developed countries .
paper-wiki-78	it is one of the most common neurological disorders in the united states , affecting approximately 10 % of the adult population .
combined	it is one of the most common brain disorders worldwide , affecting approximately 15 % of the adult population in developed countries .

4.2 Abstractive Document Summarization without Parallel Data

Originally published as: Nikolov, N. I. and Hahnloser, R. (2020). Abstractive document summarization without parallel data. In *Proceedings of the Twelfth International Conference on Language Resources and Evaluation (LREC 2020)*, Paris, France. European Language Resources Association (ELRA)

Abstract

Abstractive summarization typically relies on large collections of paired articles and summaries. However, in many cases, parallel data is scarce and costly to obtain. We develop an abstractive summarization system that relies only on large collections of example summaries and non-matching articles. Our approach consists of an unsupervised sentence extractor that selects salient sentences to include in the final summary, as well as a sentence abstractor that is trained on pseudo-parallel and synthetic data, that paraphrases each of the extracted sentences. We perform an extensive evaluation of our method: on the CNN/DailyMail benchmark, on which we compare our approach to fully supervised baselines, as well as on the novel task of automatically generating a press release from a scientific journal article, which is well suited for our system. We show promising performance on both tasks, without relying on any article-summary pairs.

4.2.1 Introduction

Text summarization aims to produce a shorter, informative version of an input text. While extractive summarization only selects important sentences from the input, abstractive summarization generates content without explicitly re-using whole sentences (Nenkova et al., 2011) resulting summaries that are more fluent. In recent years, a number of successful approaches have been proposed for both extractive (Nallapati et al., 2017; Narayan et al., 2018b) and abstractive (See et al., 2017b; Chen and Bansal, 2018b) summarization paradigms. State-of-the-art abstractive approaches are supervised, relying on large collections of paired articles and summaries. However, competitive performance of abstractive systems remains a challenge when the availability of parallel data is limited, such as in low-resource domains or for languages other than English.

Even when parallel data is severely limited, we may have access to example summaries and large collections of articles on similar topics. Examples are blog posts or scientific press releases, for which the original articles may be unavailable or behind a paywall.

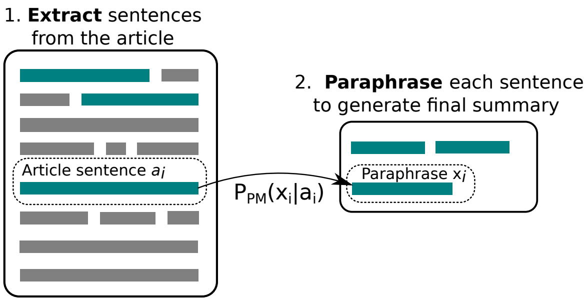


Figure 4.2: Overview of our approach to abstractive document summarization without parallel data: given an input article $\boldsymbol{a} = a_1, ..., a_N$ consisting of N sentences, (1) we fist select salient sentences a_i (in blue) using an unsupervised extractive summarization algorithm; we then (2) generate paraphrases x_i using a sentence paraphrasing model P_{PM}, trained on pseudo-parallel and synthetic data.

In this paper, we develop a system[12] for abstractive document summarization (Figure 4.2) that only relies on example summaries and non-matching articles, bypassing the need for large-scale parallel corpora. Our system consists of two components: First, an unsupervised **sentence extractor** selects salient sentences. Second, each extracted sentence is paraphrased using a **sentence abstractor**. The abstractor is trained on pseudo-parallel data extracted from raw corpora, as well as on additional synthetic data generated through backtranslation.

We evaluate our approach on two summarization tasks. First, on the CNN/-DailyMail news article summarization dataset, we compare the performance of our method based on non-parallel data to fully supervised models based on parallel data (Section 4.2.5). Second, we test our system on the novel task of automatically generating a press release from a scientific journal article (Section 4.2.6). This task is well suited for our system because only a small number of aligned document pairs can be extracted from available repositories of scientific journal articles and press releases. On both tasks, our method achieves promising results without relying on any parallel article-summary pairs.

4.2.2 Background

Supervised Summarization

In recent years, there have been large advances in supervised **abstractive summarization**, for headline generation (Rush et al., 2015; Nallapati et al., 2017) as well as for generation of multi-sentence summaries (See et al.,

[12] https://github.com/ninikolov/low_resource_summarization

2017b). State-of-the-art approaches are typically trained to generate summaries either in a fully end-to-end fashion (See et al., 2017b), processing the entire article at once, or hierarchically, first extracting content and then paraphrasing it sentence-by-sentence (Chen and Bansal, 2018b).

Both approaches rely on large collections of article-summary pairs such as the annotated Gigaword (Napoles et al., 2012) or the CNN/DailyMail dataset (Nallapati et al., 2016). The heavy reliance on manually curated resources prohibits the use of abstractive summarization in domains other than news articles, or languages other than English, where parallel data may not be as abundantly available. In such areas, extractive summarization often remains the preferred choice.

Unsupervised Summarization

Unsupervised summarization has a long history within the extractive summarization paradigm. Given an input article consisting of N sentences $\boldsymbol{a} = \{a_1, ..., a_N\}$, the goal of **extractive summarization** is to select the K most salient sentences as the output summary, without employing any paraphrasing or fusion. A typical approach is to weigh each sentence either with respect to the document as a whole (Radev et al., 2004) or through an adjacency-based measure of sentence importance (Erkan and Radev, 2004).

There is less work on unsupervised abstractive summarization. Chu and Liu (2019) propose a review summarization system based on autoencoders. Their focus is, however, on multi-document rather than on single-document summarization as it is in our case. Dohare et al. (2018) develop a pipeline for semantic abstractive summarization which works by constructing a graph from the article and then generating a summary from the most informative part of the graph. Isonuma et al. (2019) propose an abstractive summarization framework based on learning the discourse structure of input articles, and generating a single-sentence summary using a language model trained to reconstruct sentences from example reviews.

Our work focuses on multi-sentence abstractive summarization using large-scale non-parallel resources such as collections of summaries *without* matching articles. Recently, several methods have been proposed to reduce the need for parallel data either through harvesting pseudo-parallel data from raw corpora (Nikolov and Hahnloser, 2019) or by synthesizing data using backtranslation (Sennrich et al., 2016b). Such methods have been shown to be viable for a number of tasks such as unsupervised machine translation (Lample et al., 2018), sentence compression (Fevry and Phang, 2018), and style transfer (Lample et al., 2019). To the best of our knowledge, our work is the first to extend such methods to single-document summarization to generate multi-sentence abstractive summaries in a data-driven fashion.

4.2.3 Approach

Our system (see Figure 4.2) consists of two components: an **extractor** (Section 4.2.3) that picks salient sentences to include in the final summary and an **abstractor** (Section 4.2.3) that subsequently paraphrases each of the extracted sentences, rewriting them to meet the target summary style.

Our approach is similar to (Chen and Bansal, 2018b), except that they use parallel data to train their extractors and abstractors. In contrast, during training, we only assume access to M example summaries $\boldsymbol{S} = \{\boldsymbol{s}_0, .., \boldsymbol{s}_M\}$ without matching articles. During testing, given an input article consisting of N sentences $\boldsymbol{a} = \{a_0, ..., a_N\}$, our system is capable of generating a multi-sentence abstractive summary consisting of K sentences (where K is a hyperparameter).

Sentence Extractor

The extractor selects the K most salient article sentences to include in the summary. We consider two unsupervised variants for the extractor:

Lead picks the first K sentences from the article and returns them as the summary. For many datasets, such as CNN/DailyMail, Lead is a simple but tough baseline to beat, especially using abstractive methods (See et al., 2017b). Because Lead may not be the optimal choice for other domains or datasets, we experiment with another unsupervised extractive approach.

LexRank (Erkan and Radev, 2004) represents the input as a highly connected graph in which vertices represent sentences and edges between sentences are assigned weights equal to their term-frequency inverse document frequency (TF-IDF) similarity, provided that the TF-IDF similarity is higher than a predefined threshold t. The centrality of a sentence is then computed using the PageRank algorithm.

Sentence Abstractor

The sentence abstractor (P_{PM}) is trained to generate a paraphrase x_i for every article sentence a_i, rewriting it to meet the target sentence style of the summaries. We implement P_{PM} as an LSTM encoder-decoder with an attention mechanism (Bahdanau et al., 2015). Instead of training the abstractor on parallel examples of sentences from articles and summaries, we train it on a synthetic dataset created in two steps (summarized in Figure 4.3):

Pseudo-parallel dataset. The first step is to obtain an initial set of pseudo-parallel article-summary sentence pairs. Because we assume access to example summaries, our approach is to align summary sentences to an external

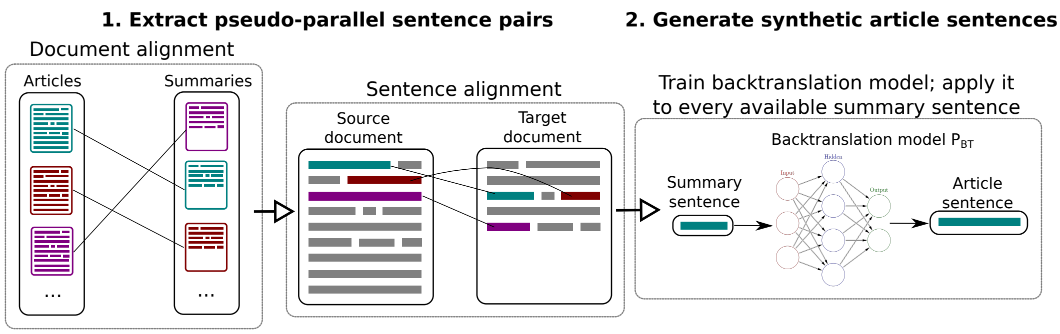


Figure 4.3: Overview of our pipeline for constructing the training dataset of our sentence abstractor P_{PM}. (1), given a dataset of summaries and articles on similar topics, we extract pseudo-parallel document and sentence pairs using large-scale alignment (Nikolov and Hahnloser, 2019). (2), we use the pseudo-parallel pairs to train a backtranslation model P_{BT}, which we use to synthesize article sentences given a summary sentence.

collection of sentences in the same format as our target summaries. Here, we apply the large-scale alignment method from (Nikolov and Hahnloser, 2019), which hierarchically aligns documents followed by sentences in the two datasets (see Figure 4.3, step 1). The alignment is implemented using nearest neighbor search: first on document and then on sentence embeddings.

Backtranslated pairs. We use the initial pseudo-parallel dataset to train a backtranslation model $P_{\text{BT}}(x_i|s_i)$, following (Sennrich et al., 2016b). The model learns to synthesize "fake" article sentences given a summary sentence (see Figure 4.3, 2). We use P_{BT} to generate multiple synthetic article sentences x_{ij} for each summary sentence s_i available, taking the $j = 1, \ldots, J$ top hypotheses predicted by beam search[13].

To train our final sentence paraphrasing model $P_{\text{PM}}(s_i|x_i)$, we combine all pseudo-parallel and backtranslated pairs into a single dataset of article-summary sentence pairs.

4.2.4 Experimental Set-up

Implementation details. P_{PM} and P_{BT} are both implemented as bidirectional LSTM encoder-decoder models with 256 hidden units, embedding dimension 128, and an attention mechanism (Bahdanau et al., 2015). We pick this model size to be comparable to recent work (See et al., 2017b; Chen and Bansal, 2018b). We set the vocabulary size to 50k and train both models

[13] We also experimented with sampling (Edunov et al., 2018) but found it to be too noisy in the current setting.

until convergence with Adam (Kingma and Ba, 2015); P_{PM} uses beam search with a beam of 5 during testing.

Because both of our extractor variants are unsupervised, we directly apply them to the articles to select salient sentences. We always set the number K of sentences to be extracted to the average number of summary sentences in the target dataset, $K = 4$ for the CNN/DailyMail dataset, and $K = 25$ for the scientific article dataset.

Evaluation details. We evaluate our systems using the ROUGE-1/2/L F1 metrics (Lin, 2004) and METEOR (Banerjee and Lavie, 2005), which is often used in machine translation.

4.2.5 Experiments on CNN/DailyMail

We use the CNN/DailyMail (CNN/DM) dataset (Hermann et al., 2015) consisting of pairs of news articles from CNN and Daily Mail, along with summaries in the form of bullet points. We choose this dataset because it allows us to compare our approach to existing fully supervised methods and to measure the gap between unsupervised and supervised summarization. We follow the preprocessing pipeline of (Chen and Bansal, 2018b), splitting the dataset into 287k/11k/11k pairs for training/validation/testing. Note that our method relies only on the bullet-point summaries from this training set.

Obtaining synthetic data. To obtain training data for our sentence abstractor P_{PM}, we follow the procedure from Section 4.2.3. We align all summaries from the CNN/DM training set to 8.5M news articles from the Gigaword dataset (Napoles et al., 2012), which *contains no articles from CNN or Daily Mail*. After alignment[14], we obtain 1.2M pseudo-parallel setnence pairs that we use to train our backtranslation model P_{BT}. Using P_{BT}, we synthesize $J = 5$ article sentences for each of the 1M summary sentences by picking the top 5 beam hypotheses. Our best sentence paraphrasing dataset used to train our final abstractor P_{PM} contains 6.7 million sentence pairs, 18% of which are pseudo-parallel pairs and 82% are backtranslated pairs.

Results on CNN/DailyMail

Baselines. We compare our models with several supervised and unsupervised baselines. LSTM is a standard bidirectional LSTM model trained to directly generate the CNN/DM summaries from the full CNN/DM articles. Ext-Abs is a hierarchical model consisting of a supervised LSTM extractor

[14] We follow the set-up from (Nikolov and Hahnloser, 2019) using the Sent2Vec embedding method (Pagliardini et al., 2018) for computing document/sentence embeddings. We use hyperparameters $\theta_d = 0.5$ and $\theta_s = \{0.60, 0.63, 0.67\}$.

Approach	R-1	R-2	R-L	MET	#
Oracle	47.33	26.43	43.69	30.76	132
Unsupervised extractive baselines					
Lead	38.78	17.57	35.49	23.67	119
LexRank	34.49	14.1	31.32	21.27	133
Supervised abstractive baselines (Trained on parallel data)					
LSTM	35.61	15.04	32.7	16.24	58
Ext-Abs†	38.38	16.12	36.04	19.39	
Ext-Abs-RL†	40.88	17.8	38.54	20.38	73
Abstractive summarization without parallel data (this work)					
Lead + $\text{Abs}_{\text{pp+syn-5}}$	32.98	11.13	30.88	13.51	50
LexRank + $\text{Abs}_{\text{pp+syn-5}}$	30.87	9.42	28.82	12.51	52

Table 4.15: Metric results on the CNN/Daily Mail test set. **R-1/2/L** are the **ROUGE-1/2/L** F1 scores; **MET** is **METEOR**, while **#** is the average number of tokens in the summaries. † are from Chen and Bansal (2018b).

(1) $\text{Abs}_{\text{pp-0.63}}$: cnn is the first time in three years . the other contestants told the price of the price . the game show will be hosted by the tv game show . the game of the game is the first of the show .
(2) $\text{Abs}_{\text{pp+syn-5}}$: a tv legend has returned to the first time in eight years . contestants told the price of “ the price is right ” bob barker hosted the tv game show for 35 years . the game is the first of the show ’s “ lucky seven ”
(3) Abs_{par}: a tv legend returned to doing what he does best . contestants told to “ come on down ! ” on april 1 edition . he hosted the tv game show for 35 years before stepping down in 2007 . barker handled the first price-guessing game of the show , the classic “ lucky seven ”

Table 4.16: Example outputs on the CNN/DM dataset (Lead extractor): (1)/(2) are trained on pseudo-parallel/synthetic data, while the abstractor in (3) is trained on parallel data.

Approach (# pairs)	R-1	R-2	R-L	#
$\text{Abs}_{\text{pp-0.60}}$ (2M)	23.08	4.06	21.48	62
$\text{Abs}_{\text{pp-0.63}}$ (1.2M)	28.08	7.07	26.14	49
$\text{Abs}_{\text{pp-0.67}}$ (0.3M)	24.36	4.76	22.64	57
$\text{Abs}_{\text{pp+syn-1}}$ (2.4M)	31.92	10.2	29.9	51
$\text{Abs}_{\text{pp+syn-5}}$ (6.6M)	32.98	11.13	30.88	50
$\text{Abs}_{\text{pp+syn-10}}$ (12M)	32.8	11.2	30.71	49
$\text{Abs}_{\text{pp-ub}}$ (575K)	38.42	15.98	35.8	65
Abs_{par} (1M)	38.68	16.36	36.15	62

Table 4.17: Comparison of abstractors trained on parallel (Abs_{par}) vs. pseudo-parallel data ($\text{Abs}_{\text{pp-}\theta_s}$, using different sentence alignment thresholds θ_s; $\text{Abs}_{\text{pp-ub}}$ is the upper bound for large-scale alignment) or using a mixture of pseudo-parallel and synthetic data ($\text{Abs}_{\text{pp+syn-N}}$, using the $\text{Abs}_{\text{pp-0.63}}$ dataset and backtranslated data from the top N beam hypotheses). We always use the Lead extractor.

and separate abstractor (Chen and Bansal, 2018b), both of which are individually trained on the CNN/DM dataset by aligning summary to article sentences. Our work best resembles Ext-Abs except that we do not rely on any parallel data. Ext-Abs-RL is a state-of-the-art summarization system that extends Ext-Abs by jointly tuning the two supervised components using reinforcement learning. We additionally report the performance of our unsupervised extractive baselines, Lead, and LexRank, as well as the result of an oracle (Oracle) which computes an upper bound for extractive summarization by aligning the ground truth summary sentences to their original articles using ROUGE-1.

Automatic evaluation. Our best abstractive models trained on non-parallel data (Lead + $\text{Abs}_{\text{pp+syn-5}}$ and LexRank + $\text{Abs}_{\text{pp+syn-5}}$ in Table 4.15) performed worse than the baselines trained on parallel data. However, the results are promising: for example, the ROUGE-L gap between our Lead model and the supervised LSTM model is only 1.8. When comparing against the Ext-Abs and Ext-Abs-RL models, which perform supervised sentence extraction followed by supervised sentence abstraction, the gap is larger. Furthermore, we observe that, on this dataset, applying our abstractors to the Lead and LexRank extractive baselines leads to a decrease in ROUGE. This could be due to the much shorter length of our abstractive summaries in comparison to the length produced by other systems, indicating that our systems potentially summarize much more aggressively.

Model analysis. In Table 4.17, we compare the effect of training our abstractor on pseudo-parallel datasets of different sizes ($\text{Abs}_{\text{pp-*}}$) as well as on a mixture of pseudo-parallel and backtranslated data ($\text{Abs}_{\text{pp+syn-*}}$). For reference, we also include results from aligning the original dataset of CNN/DM articles and summaries directly. We construct a *parallel* dataset (Abs_{par}) of sentence pairs by aligning the original CNN/DM document pairs using ROUGE-1; as well as a pseudo-parallel dataset ($\text{Abs}_{\text{pp-ub}}$) by applying the large-scale alignment method to the CNN/DM documents, (without using the document labels directly). The performance difference between Abs_{par} and $\text{Abs}_{\text{pp-ub}}$ provides an estimate of the performance loss due to pseudo alignment.

Our best pseudo-parallel abstractor performs poorly in comparison to the parallel abstractor. Adding additional synthetic data is helpful but insufficient to compensate for the performance gap. Furthermore, we observe a diminishing improvement from adding synthetic pairs. By contrast, the large-scale alignment method constructs a pseudo-parallel upper bound that almost perfectly matches the parallel dataset, indicating that potentially the main bottleneck in our system is the domain difference between the articles in Gigaword and the CNN/DM.

Example summaries. In Table 4.16, we also provide example summaries produced on the CNN/DM dataset. Our final model trained on additional backtranslated data produced much more relevant and coherent sentences than the model trained on pseudo-parallel data only. Despite having seen no parallel examples, the system is capable of generating fluent, abstractive sentences. However, in comparison to the abstractor trained on parallel data, there is still room for further improvement.

4.2.6 Experiments on Scientific Articles

The rate of scientific publications grows exponentially (Hunter and Cohen, 2006), calling for efficient automatic summarization tools (Nikolov et al., 2018a). An important frontier in scientific text summarization is generating summaries that not only synthesize an article but make it also more accessible to non-specialists (Vadapalli et al., 2018a,b; Tatalovic, 2018). A major challenge towards achieving this goal is the lack of parallel datasets of articles and high-quality summaries.

We introduce the novel task of automatically generating a press release for a scientific article. Although there are already large repositories of scientific articles and press releases, two major obstacles are preventing manual alignment across these resources:

1. Because press releases address very different audiences, they are often written and published separately from the article. Furthermore, there is often no metadata present in the text (such as a digital object iden-

Table 4.18: Datasets used to extract pseudo-parallel monolingual sentence pairs in our style transfer experiments.

Dataset	Documents	Tok. per sent.	Sent. per doc.
PubMed	1.5M	24 ± 14	180 ± 98
MEDLINE	16.8M	26 ± 13	7 ± 4
EurekAlert	358K	29 ± 15	25 ± 13
Wikipedia	5.5M	25 ± 16	17 ± 32

tifier (DOI)) that could be used to link a press release to its original scientific article.

2. Even when there is metadata that can be utilized, the full text of the original article may not be accessible, either because it is published in a closed access journal under a restrictive license, or because the full text of the article is not available in an easily parsable format (e.g., only a PDF is available).

These practical challenges call for alternative summarization approaches. The summarization method described in this paper circumvents these problems because it exploits the large existing collections of papers and press releases that exist within open repositories.

Datasets used for alignment. To extract pseudo-parallel sentence pairs for paraphrasing, we rely on four collections of documents. We combine two datasets of scientific articles: `PubMed`[15], which contains the full text of $1.5M$ open access papers, and `Medline`[16], which contains over $17M$ scientific abstracts. We obtain press releases through scraping $350k$ articles from Eurekalert[17], an aggregator that covers many scientific disciplines. As an additional resource that contains scientific texts written for a more general audience, we use all articles on Wikipedia, without applying any filtering. Table 4.18 contains an overview of these datasets.

Evaluation dataset. To create a parallel testing dataset, we use regular expressions to detect digital object identifier (DOI) mentions within the press releases. Given a DOI, we query for the full text of a paper using the Elsevier ScienceDirect API[18]. Using this approach, we were able to compose 6821 parallel pairs of the full text of a paper and its press release (which amounts to only 2% of all press releases available). We use these 6821 pairs

[15] `www.ncbi.nlm.nih.gov/pmc/tools/openftlist`
[16] `www.nlm.nih.gov/bsd/medline.html`
[17] `https://www.eurekalert.org/`
[18] `https://dev.elsevier.com/sd_apis.html`

Approach	R-1	R-2	R-L	MET	#
Oracle	43.79	14.27	41.1	20.05	969
Unsupervised extractive baselines					
Lead	41.2	11.85	38.87	17.11	688
LexRank	38.78	10.31	36.57	16.66	800
Abstractive summarization without parallel data (this work)					
Lead + $\text{Abs}_{\text{pp+syn-1}}$	42.47	12.11	40.25	15.78	529
LexRank + $\text{Abs}_{\text{pp+syn-1}}$	41.04	10.94	38.9	15.38	562

Table 4.19: Metric results on the Scientific summarization test set. **R-1/2/L** are the **ROUGE-1/2/L** F1 scores; **MET** is **METEOR**, while **#** is the average number of tokens in the summaries.

as our testing set in our experiments, and exclude them from the alignment procedure.

Extracting pseudo-parallel data. To obtain a pseudo-parallel dataset for scientific sentence paraphrasing, we aligned the scientific papers to the press releases. After alignment[19], we extracted 80*k* pseudo-parallel pairs in total. We additionally aligned `PubMed` and `Medline` to all articles on `Wikipedia`[20], obtaining out-of-domain pairs for this task. We merged all pairs into a single dataset, consisting of 370*k* sentence pairs. We used these pairs to train a backtranslation model P_{BT} from which we synthesized 1 article sentence for each sentence in the press release dataset[21]. Our final sentence paraphrasing dataset used to train the abstractor P_{PM} contains 8.6*M* pairs.

Results

Baselines. Since there are no existing parallel datasets of scientific articles and press releases, here we do not report results using supervised methods. We compare the performance of our models (Lead + $\text{Abs}_{\text{pp+syn-1}}$ and LexRank + $\text{Abs}_{\text{pp+syn-1}}$) to the two unsupervised extractive baselines, Lead and LexRank respectively, as well as to the result of the Oracle (same as in Section 4.2.5) that yields an upper bound for extractive summarization.

Model analysis. Our results are in Table 4.19. Both of our abstractive models outperformed their respective extractive baselines, indicating that our abstractors are beneficial for this task and have learned useful sentence trans-

[19] We use a document similarity threshold $\theta_d = 0.6$ and a sentence similarity threshold $\theta_s = 0.74$.

[20] Using $\theta_d = 0.6$ and $\theta_s = 0.78$.

[21] We synthesize only a single sentence because the press release dataset contains many more sentences than CNN/DailyMail.

Lead: global reference models , such as preliminary reference earth model (prem ; dziewonski and anderson , 1981) , iasp91 (kennett and engdahl , 1991) and ak135 (kennett et al. , 1995) were created with different data sets , techniques and assumptions , leading to differences in some regions of earth .
Lead + $\text{Abs}_{\text{pp+syn-1}}$: the study was created with different data sets , techniques and assumptions , leading to differences in some regions of earth .
Lead: astroglia from the postnatal cerebral cortex can be reprogrammed in vitro to generate neurons following forced expression of neurogenic transcription factors , thus opening new avenues towards a potential use of endogenous astroglia for brain repair .
Lead + $\text{Abs}_{\text{pp+syn-1}}$: the brain ’ s neural network has been reprogrammed to generate neurons from the postnatal cerebral cortex .
Lead: sequenced the angiotensinogen gene and found that multiple rare variants contribute to variation in angiotensinogen levels ; interestingly , most of these variants sit on the same haplotype background created by three common snps.26 fourth , three common tag snps encompassing mc1r (mim 155555) were independently associated with melanoma in a recent gwas.27 however , resequencing of the candidate gene , mc1r , indicates that these signals can be completely explained by the combined effects of several rare nonsynonymous mutations , suggesting that ignoring rare variants can lead to incorrect inferences on the potential role of candidate genes carrying common snps identified by gwas (f. demenais at al .
Lead + $\text{Abs}_{\text{pp+syn-1}}$: the researchers found that these variants can lead to incorrect inferences on the potential role of candidate genes carrying common snps .

Table 4.20: Example sentence paraphrases produced on the scientific dataset (using the Lead extractor).

formations. The summaries are much shorter after abstraction, indicating that the paraphrased summaries are meaningfully compressed. Surprisingly, the gap between the baselines and the oracle is small, indicating that abstractive summarization is essential for achieving good performance on this task.

Example summaries. We find that the model trained on backtranslated data is often conservative choosing to leave many input sentences almost unchanged. However, as shown by examples in Table 4.20, the model learned useful transformations by compressing long sentences and utilizing vocabulary adapted to the language of press releases (e.g. ”the researchers” in the last example). The sentence compressions can sometimes seem too drastic when looked at in isolation (e.g., from ”astroglia from the postnatal cerebral cortex” to ”the brain’s neural network” in the second example).

4.2.7 Conclusion

We developed an abstractive summarization system that does not rely on parallel resources, but can instead be trained using example summaries and a large collection of non-matching articles, making it particularly relevant to low-resource domains and languages. On the CNN/DailyMail benchmark, our system performed competitively to fully supervised models. We also achieved promising results on the novel task of automatically generating a press release for a scientific journal article.

Future work will focus on developing novel unsupervised extractors, on decreasing the gap between abstractors trained on parallel and non-parallel data, as well as on developing methods for combining the abstractor and extractor into a single system.

Chapter 5

Conclusion and Future Work

This thesis explores abstractive document summarization from two different perspectives. First, in Chapter 3, we considered data-driven methods that are trained on large collections of parallel examples. Second, in Chapter 4, we considered methods which are still data-driven but circumvent the requirement for parallel data through the construction of pseudo-parallel and synthetic datasets. This chapter summarizes the main findings of the thesis and discusses relevant directions for future work.

5.1 Data-driven Summarization

5.1.1 Summarization of very long articles

The majority of past work in text modeling and generation focuses on processing short sequences, such as individual sentences. For some tasks, such as summarization or machine translation, working on the sentence level might be insufficient since valuable contextual information may be present in adjacent sentences. Therefore, an important research frontier is to scale text generation models to whole documents (Läubli et al., 2018) and potentially even to whole books[1].

In Chapter 3.1, we considered two novel text generation datasets which contain long sequences: generating the title of a scientific paper from its abstract, or its abstract from the full body. While the results for title generation were promising, the models we considered at the time the paper was written struggled with generating the abstract.

Our results highlight that although many modern sequence models can perform very well on the sentence level, often they face challenges when trained

[1] https://www.springer.com/gp/about-springer/media/press-releases/corporate/springer-nature-machine-generated-book/16590126

on very long input or output sequences. Two main difficulties need to be addressed when developing novel models that can process long texts better.

First, on a more practical level, modeling/generation of long documents is very computationally demanding, requiring longer training times and GPUs with large memory capacity. Recent work in (Radford et al., 2019) has shown that very large language models (e.g., containing over 1 billion parameters) have the capacity to generate long samples with quality that is far superior to that of smaller architectures. Training such large models, however, is very time-consuming and resource-demanding, making them prohibitive to use with standard off-the-shelf GPU hardware. An exciting frontier is to develop more compact architecture variants that can perform on par with these very large models. Efforts are underway to develop more efficient models, for example, through working at reduced bit precision (Hubara et al., 2017), or by optimizing architectures to reduce parameter redundancy (Kitaev et al., 2020). Another exciting direction is to use neural architecture search (So et al., 2019) to optimize the models for efficiency.

Second, working with longer sequences may require hierarchical processing of the text. Novel hierarchical architectures need to be capable of preserving long-term dependencies while maintaining coherence in the generated text. In automatic summarization, hybrid extractive-abstractive models such as (Chen and Bansal, 2018a) have been gaining popularity. Chen and Bansal (2018a) first perform an extraction step followed by abstractive paraphrasing, both steps learned by separate models that are independently trained. Hybrid approaches offer a good trade-off between output summary quality and computational efficiency since text generation is typically performed on the sentence level. However, since the output sentences are generated independently, this could lead to lower coherence across sentences.

Since our work, a number of other studies have focused on summarization of scientific articles, and summarization of long documents in general. In the concurrent work of Cohan et al. (2018), the authors propose a novel model that specifically targets summarization of long documents. They test the model on scientific summarization, constructing two datasets for generating the abstract of a paper, using papers from arXiv and PubMed. The main difference between our work and theirs is that our datasets are substantially larger (900k article-summary pairs for abstract generation, vs. $\approx$200k pairs for their datasets). Furthermore, we consider the additional task of generating the title of a paper from its abstract, for which we construct a dataset that contains over 5 million pairs, and we also benchmark a wide range of extractive and abstractive systems not considered by Cohan et al. (2018). In other work, Wang et al. (2018) develop an abstract generation model that performs text editing rather than full generation from scratch. They test their model using a dataset of 10k papers collected from the ACL anthology. Liu* et al. (2018) develop a Transformer model that can generate long summaries of Wikipedia articles.

5.1.2 Summary Denoising

In Chapter 3.2, we considered the problem of automatically improving machine-generated text as a postprocessing step. Our work is related to several more recent studies that use self-supervised noise objectives, such as shuffling or token masking, to pre-train sequence models (Raffel et al., 2019; Lewis et al., 2019).

Similarly, our method applies a noise function to a dataset of summaries to construct datasets for summary refinement. We consider several different noise types that focus specifically on repetition errors that are common in summarization. Since our method acts as a postprocessing step, rather than a pre-training step, it is independent of the type of the summarization system being used, or whether the system performs extractive or abstractive summarization. Our summary refiniement approach worked particularly well for extractive summarization, where it proved to be effective at removing redundant information from the summaries, leading to ROUGE improvements. Denoising abstractive summarization proved to be more challenging, and we were not able to show ROUGE improvements over a state-of-the-art system. However, our analysis in Figure 3.6 shows that even state-of-the-art abstractive systems produce summaries that contain redundant information, motivating further research in the area.

Future work on summary denoising should focus on developing novel noise functions that target other errors common in machine-generated abstractive text. Three potential candidates are grammatical errors, low coherence in a multi-sentence text, or incorrect order of the presented information. Specialized automatic metrics should also be developed, that can detect such errors in the generated summaries, and that can quantify their impact across multiple summaries.

5.2 Paraphrasing and Summarization without Parallel Data

5.2.1 Large-scale alignment

In Chapter 4.1, we moved away from methods that rely on parallel training data and focused on the problem of text rewriting *only with raw, non-parallel resources*. We developed a simple large-scale alignment method capable of extracting pseudo-parallel sentence pairs from two comparable monolingual corpora composed of different text styles. Our approach works by hierarchically searching for similar documents, and subsequently, similar sentences across the two comparable corpora.

We evaluated the performance of our method on automatic alignment benchmarks, as well as extrinsically, on the task of automatic text simplification. We also performed a human evaluation to assess the quality of our simplification models. We reported improvements arising from adding pseudo-parallel sentence pairs to existing parallel datasets, as well as promising results when using the pseudo-parallel data on its own. Our results demonstrate that careful engineering of pseudo-parallel datasets can be a successful approach for improving existing monolingual text-to-text rewriting tasks, as well as for tackling novel tasks for which little or no parallel data is currently available.

One of the limitations of our approach is that it relies on high-quality pseudo-parallel resources, which may not always be available for all tasks or all languages. Future work could consider investigating transfer learning approaches for models specially trained on pseudo-parallel data, which could help to address this challenge. Another limitation is that our approach may be too unconstrained for more specific use cases, such as for training sentence compression models. Addressing this would require developing more sophisticated methods for sentence filtering that apply more constraints during the final pair selection process.

Finally, due to our open problem definition, we have no rigorous way to measure and control the amount of noise present in the output pseudo-parallel datasets. To some extent, this is addressed by the similarity thresholds for acceptance, which we introduce. However, in practice, picking the thresholds can be challenging, and their values may vary depending on the domains and tasks under consideration. Future work should, therefore, explore more sophisticated unsupervised strategies to sentence pairing and final pair selection, that can better filter high-quality pairs from a large number of noisy pairs extracted by our system.

5.2.2 Abstractive Summarization without Parallel Data

In Chapter 4.2, we used our large-scale alignment system as a basis for developing abstractive summarization systems that do not rely on any parallel resources but can instead be trained using a collection of example target summaries, as well as a large collection of non-matching raw articles. We tested our method on two summarization tasks: on the standard CNN/-DailyMail news summarization benchmark (Hermann et al., 2015; See et al., 2017a), as well as on the novel task of automatically generating a press release for a scientific journal article. The latter is particularly well suited for our system since there are large existing collections of scientific papers and press releases. However, only a few press releases contain links back to their originating scientific article, prohibiting the automated extraction of parallel training pairs.

On CNN/DailyMail, our abstractive summarization system performed competitively to fully supervised models, while on the scientific dataset, we outperformed several strong unsupervised extractive baselines. Our results demonstrate the potential of our low-resource summarization approach.

One of the main limitations of our system is that it assumes *some* example summaries are available. Often, this may not be the case, especially when considering summarization for low-resource languages other than English. Therefore, an important future direction is to develop alternative approaches to low-resource sentence abstraction that relax this requirement.

Another limitation is that our summarization approach currently consists of two disjoint steps, sentence extraction, and paraphrasing. Since the two steps are performed independently, there is no guarantee that an output summary will be coherent across all of its sentences. An important future direction is, therefore, to integrate the extractor and abstractor components into a single end-to-end system or to at least develop approaches for fine-tuning the two components jointly, similar to Chen and Bansal (2018a). The latter is particularly challenging due to our assumption that there is no parallel training data available. Two additional promising future directions that may result in improvements to our system are to use self-supervised objectives to pre-train our abstractors and extractors, similar to (Zhang et al., 2019), and to test more recent unsupervised extractive summarization approaches such as (Zheng and Lapata, 2019; West et al., 2019).

Bibliography

Abrahamsson, E., Forni, T., Skeppstedt, M., and Kvist, M. (2014). Medical text simplification using synonym replacement: Adapting assessment of word difficulty to a compounding language. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, pages 57–65.

Adiwardana, D., Luong, M.-T., So, D. R., Hall, J., Fiedel, N., Thoppilan, R., Yang, Z., Kulshreshtha, A., Nemade, G., Lu, Y., et al. (2020). Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Alva-Manchego, F., Scarton, C., and Specia, L. (2020). Data-driven sentence simplification: Survey and benchmark. *Computational Linguistics*, (Just Accepted):1–87.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Banerjee, S. and Lavie, A. (2005). Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proc. of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Barzilay, R. and Elhadad, N. (2003). Sentence alignment for monolingual comparable corpora. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 25–32. Association for Computational Linguistics.

Biber, D. and Gray, B. (2010). Challenging stereotypes about academic writing: Complexity, elaboration, explicitness. *Journal of English for Academic Purposes*, 9(1):2–20.

Björk, B.-C., Roos, A., and Lauri, M. (2008). Global annual volume of peer reviewed scholarly articles and the share available via different open access options. In *ELPUB2008*.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Brokos, G.-I., Malakasiotis, P., and Androutsopoulos, I. (2016). Using centroids of word embeddings and word mover's distance for biomedical document retrieval in question answering. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, pages 114–118, Berlin, Germany. Association for Computational Linguistics.

Brown, P. F., Lai, J. C., and Mercer, R. L. (1991). Aligning sentences in parallel corpora. In *Proceedings of the 29th annual meeting on Association for Computational Linguistics*, pages 169–176. Association for Computational Linguistics.

Cambria, E. and White, B. (2014). Jumping NLP curves: a review of natural language processing research [review article]. *Computational Intelligence Magazine, IEEE*, 9(2):48–57.

Castilho, S., Moorkens, J., Gaspari, F., Calixto, I., Tinsley, J., and Way, A. (2017). Is neural machine translation the new state of the art? *The Prague Bulletin of Mathematical Linguistics*, 108(1):109–120.

Chatterjee, R., Negri, M., Rubino, R., and Turchi, M. (2018). Findings of the wmt 2018 shared task on automatic post-editing. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 710–725.

Chen, Y.-C. and Bansal, M. (2018a). Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–686, Melbourne, Australia. Association for Computational Linguistics.

Chen, Y.-C. and Bansal, M. (2018b). Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of ACL*.

Chiu, B., Crichton, G., Korhonen, A., and Pyysalo, S. (2016). How to train good word embeddings for biomedical nlp. *Proceedings of BioNLP16*, page 166.

Cho, K., Courville, A., and Bengio, Y. (2015). Describing multimedia content using attention-based encoder-decoder networks. *IEEE Transactions on Multimedia*, 17(11):1875–1886.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Chopra, S., Auli, M., Rush, A. M., and Harvard, S. (2016). Abstractive sentence summarization with attentive recurrent neural networks. *NAACL 2016*.

Chu, E. and Liu, P. (2019). Meansum: A neural model for unsupervised multi-document abstractive summarization. In *International Conference on Machine Learning*, pages 1223–1232.

Cohan, A., Dernoncourt, F., Kim, D. S., Bui, T., Kim, S., Chang, W., and Goharian, N. (2018). A discourse-aware attention model for abstractive summarization of long documents. 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 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Cohan, A. and Goharian, N. (2018). Scientific document summarization via citation contextualization and scientific discourse. *International Journal on Digital Libraries*, 19(2-3):287–303.

Collins, E., Augenstein, I., and Riedel, S. (2017). A supervised approach to extractive summarisation of scientific papers. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 195–205, Vancouver, Canada. Association for Computational Linguistics.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. *Proc. of EMNLP 2017*.

Coster, W. and Kauchak, D. (2011). Simple english wikipedia: a new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 665–669. Association for Computational Linguistics.

Denkowski, M. and Lavie, A. (2014). Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). 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. Association for Computational Linguistics.

Dietterich, T. G., Domingos, P., Getoor, L., Muggleton, S., and Tadepalli, P. (2008). Structured machine learning: the next ten years. *Machine Learning*, 73(1):3–23.

Dohare, S., Gupta, V., and Karnick, H. (2018). Unsupervised semantic abstractive summarization. In *Proceedings of ACL 2018, Student Research Workshop*, pages 74–83.

Edunov, S., Ott, M., Auli, M., and Grangier, D. (2018). Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500.

Erkan, G. and Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

Fee, M. S. (2012). Oculomotor learning revisited: a model of reinforcement learning in the basal ganglia incorporating an efference copy of motor actions. *Frontiers in neural circuits*, 6:38.

Fevry, T. and Phang, J. (2018). Unsupervised sentence compression using denoising auto-encoders. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 413–422. Association for Computational Linguistics.

Filippova, K., Alfonseca, E., Colmenares, C. A., Kaiser, L., and Vinyals, O. (2015). Sentence compression by deletion with LSTMs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 360–368, Lisbon, Portugal. Association for Computational Linguistics.

Fu, Z., Tan, X., Peng, N., Zhao, D., and Yan, R. (2017). Style transfer in text: Exploration and evaluation. *arXiv preprint arXiv:1711.06861*.

Ganitkevitch, J., Van Durme, B., and Callison-Burch, C. (2013). Ppdb: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764.

Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1243–1252. JMLR.org.

Gehrmann, S., Deng, Y., and Rush, A. (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.

Giridharan, V. V., Thandavarayan, R. A., Arumugam, S., Mizuno, M., Nawa, H., Suzuki, K., Ko, K. M., Krishnamurthy, P., Watanabe, K., and Konishi, T. (2015). Schisandrin b ameliorates icv-infused amyloid β induced oxidative stress and neuronal dysfunction through inhibiting rage/nf-κb/mapk and up-regulating hsp/beclin expression. *PLoS One*, 10(11):e0142483.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Grégoire, F. and Langlais, P. (2018). Extracting parallel sentences with bidirectional recurrent neural networks to improve machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1442–1453, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Hardy, H., Narayan, S., and Vlachos, A. (2019). HighRES: Highlight-based reference-less evaluation of summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3381–3392, Florence, Italy. Association for Computational Linguistics.

Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701.

Hirschberg, J. and Manning, C. D. (2015). Advances in natural language processing. *Science*, 349(6245):261–266.

Hoffman, M., Bach, F. R., and Blei, D. M. (2010). Online learning for latent dirichlet allocation. In *advances in neural information processing systems*, pages 856–864.

Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., and Bengio, Y. (2017). Quantized neural networks: Training neural networks with low precision weights and activations. *J. Mach. Learn. Res.*, 18(1):6869–6898.

Hunter, L. and Cohen, K. B. (2006). Biomedical language processing: what's beyond pubmed? *Molecular cell*, 21(5):589–594.

Hwang, W., Hajishirzi, H., Ostendorf, M., and Wu, W. (2015). Aligning sentences from standard wikipedia to simple wikipedia. In *HLT-NAACL*, pages 211–217.

Inui, K., Fujita, A., Takahashi, T., Iida, R., and Iwakura, T. (2003). Text simplification for reading assistance: a project note. In *Proceedings of the second international workshop on Paraphrasing-Volume 16*, pages 9–16. Association for Computational Linguistics.

Isabelle, P. (2007). Domain adaptation of mt systems through automatic postediting. *Proc. 10th Machine Translation Summit (MT Summit XI), 2007*.

Isonuma, M., Mori, J., and Sakata, I. (2019). Unsupervised neural single-document summarization of reviews via learning latent discourse structure and its ranking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2142–2152, Florence, Italy. Association for Computational Linguistics.

Jurafsky, D. and Martin, J. H. (2008). Speech and language processing: An introduction to speech recognition, computational linguistics and natural language processing. *Upper Saddle River, NJ: Prentice Hall*.

Kajiwara, T. and Komachi, M. (2016). Building a monolingual parallel corpus for text simplification using sentence similarity based on alignment between word embeddings. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1147–1158.

Kalchbrenner, N., Espeholt, L., Simonyan, K., van den Oord, A., Graves, A., and Kavukcuoglu, K. (2016). Neural machine translation in linear time. *CoRR*, abs/1610.10099.

Kim, M., Moirangthem, D. S., and Lee, M. (2016). Towards abstraction from extraction: Multiple timescale gated recurrent unit for summarization. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 70–77, Berlin, Germany. Association for Computational Linguistics.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Kitaev, N., Kaiser, L., and Levskaya, A. (2020). Reformer: The efficient transformer. In *International Conference on Learning Representations*.

Knight, K. and Marcu, D. (2002). Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*, 139(1):91–107.

Koehn, P. (2009). *Statistical machine translation*. Cambridge University Press.

Koumpis, K. and Renals, S. (2005). Automatic summarization of voicemail messages using lexical and prosodic features. *ACM Trans. Speech Lang. Process.*, 2(1):1–es.

Kusner, M., Sun, Y., Kolkin, N., and Weinberger, K. (2015). From word embeddings to document distances. In *International conference on machine learning*, pages 957–966.

Lample, G., Ott, M., Conneau, A., Denoyer, L., and Ranzato, M. (2018). Phrase-based & neural unsupervised machine translation. In *Proceedings* *of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049, Brussels, Belgium. Association for Computational Linguistics.

Lample, G., Subramanian, S., Smith, E., Denoyer, L., Ranzato, M., and Boureau, Y.-L. (2019). Multiple-attribute text rewriting. In *International Conference on Learning Representations*.

Läubli, S., Sennrich, R., and Volk, M. (2018). Has machine translation achieved human parity? a case for document-level evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4791–4796, Brussels, Belgium. Association for Computational Linguistics.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196.

Lebanoff, L., Song, K., and Liu, F. (2018). Adapting the neural encoder-decoder framework from single to multi-document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4131–4141, Brussels, Belgium. Association for Computational Linguistics.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lee, J., Cho, K., and Hofmann, T. (2017). Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, 5:365–378.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of psychology*.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8.

Lin, C.-Y. and Hovy, E. (2002). From single to multi-document summarization. In *Proceedings of the 40th annual meeting of the association for computational linguistics*, pages 457–464.

Liu, G., Rosello, P., and Sebastian, E. (2016). Style transfer with non-parallel corpora.

Liu*, P. J., Saleh*, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., and Shazeer, N. (2018). Generating wikipedia by summarizing long sequences. In *International Conference on Learning Representations*.

Lloret, E., Romá-Ferri, M. T., and Palomar, M. (2013). Compendium: A text summarization system for generating abstracts of research papers. *Data & Knowledge Engineering*, 88:164–175.

Mallinson, J., Severyn, A., Malmi, E., and Garrido, G. (2020). Felix: Flexible text editing through tagging and insertion. *arXiv preprint arXiv:2003.10687*.

Malmi, E., Krause, S., Rothe, S., Mirylenka, D., and Severyn, A. (2019). Encode, tag, realize: High-precision text editing. 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 5054–5065, Hong Kong, China. Association for Computational Linguistics.

Mamoshina, P., Vieira, A., Putin, E., and Zhavoronkov, A. (2016). Applications of deep learning in biomedicine. *Molecular pharmaceutics*, 13(5):1445–1454.

Mani, I. (2001). *Automatic summarization*, volume 3. John Benjamins Publishing.

Marie, B. and Fujita, A. (2017). Efficient extraction of pseudo-parallel sentences from raw monolingual data using word embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 392–398.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *Proceedings of Workshop at ICLR*, 2013.

Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In *INTERSPEECH*, volume 2, page 3.

Moore, R. C. (2002). Fast and accurate sentence alignment of bilingual corpora. In *Conference of the Association for Machine Translation in the Americas*, pages 135–144. Springer.

Munteanu, D. S. and Marcu, D. (2005). Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4):477–504.

Nallapati, R., Zhai, F., and Zhou, B. (2017). Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Nallapati, R., Zhou, B., dos Santos, C., Gulçehre, 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. Association for Computational Linguistics.

Napoles, C., Callison-Burch, C., and Post, M. (2016). Sentential paraphrasing as black-box machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 62–66, San Diego, California. 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*, pages 95–100. Association for Computational Linguistics.

Narayan, S., Cohen, S. B., and Lapata, M. (2018a). 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. Association for Computational Linguistics.

Narayan, S., Cohen, S. B., and Lapata, M. (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. Association for Computational Linguistics.

Nenkova, A. and Bagga, A. (2004). Facilitating email thread access by extractive summary generation. In *Proceedings of the 2003 International Conference on Recent Advances in Natural Language Processing (RANLP)*, pages 287–296.

Nenkova, A., Maskey, S., and Liu, Y. (2011). Automatic summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts of ACL 2011*, page 3. 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. Association for Computational Linguistics.

Nikolov, N., Pfeiffer, M., and Hahnloser, R. (2018a). Data-driven summarization of scientific articles. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

Nikolov, N. I., Calmanovici, A., and Hahnloser, R. (2019). Summary refinement through denoising. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2019*.

Nikolov, N. I. and Hahnloser, R. (2019). Large-scale hierarchical alignment for data-driven text rewriting. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2019*.

Nikolov, N. I. and Hahnloser, R. (2020). Abstractive document summarization without parallel data. In *Proceedings of the Twelfth International Conference on Language Resources and Evaluation (LREC 2020)*, Paris, France. European Language Resources Association (ELRA).

Nikolov, N. I., Hu, Y., Tan, M. X., and Hahnloser, R. (2018b). Character-level Chinese-English Translation through ASCII Encoding. In *Third Conference on Machine Translation (WMT18)*, Brussels, Belgium.

Nikolov, N. I., Pfeiffer, M., and Hahnloser, R. (2018c). Data-driven summarization of scientific articles. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

Nisioi, S., Štajner, S., Ponzetto, S. P., and Dinu, L. P. (2017). Exploring neural text simplification models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 85–91.

Ouyang, J., Song, B., and McKeown, K. (2019). 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. 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.

Pagliardini, M., Gupta, P., and Jaggi, M. (2018). Unsupervised learning of sentence embeddings using compositional n-gram features. 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 528–540. Association for Computational Linguistics.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th* *annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Parida, S. and Motlicek, P. (2019). Abstract text summarization: A low resource challenge. 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 5996–6000.

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

Plavén-Sigray, P., Matheson, G. J., Schiffler, B. C., and Thompson, W. H. (2017). The readability of scientific texts is decreasing over time. *Elife*, 6:e27725.

Prabhumoye, S., Tsvetkov, Y., Salakhutdinov, R., and Black, A. W. (2018). Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–876, Melbourne, Australia. Association for Computational Linguistics.

Qazvinian, V. and Radev, D. R. (2008). Scientific paper summarization using citation summary networks. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 689–696. Association for Computational Linguistics.

Radev, D. R., Jing, H., Styś, M., and Tam, D. (2004). Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Robertson, S., Zaragoza, H., et al. (2009). The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Rossiello, G., Basile, P., and Semeraro, G. (2017). Centroid-based text summarization through compositionality of word embeddings. *MultiLing 2017*, page 12.

Rozovskaya, A. and Roth, D. (2016). Grammatical error correction: Machine translation and classifiers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2205–2215.

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. Association for Computational Linguistics.

See, A., Liu, P. J., and Manning, C. D. (2017a). 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.

See, A., Liu, P. J., and Manning, C. D. (2017b). 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)*, volume 1, pages 1073–1083.

Sennrich, R., Haddow, B., and Birch, A. (2016a). 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. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016b). Improving neural machine translation models with monolingual data. In *Proc. of ACL*, pages 86–96. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016c). 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. Association for Computational Linguistics.

Serban, I. V., Sordoni, A., Bengio, Y., Courville, A., and Pineau, J. (2016). Building end-to-end dialogue systems using generative hierarchical neural network models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Shen, T., Lei, T., Barzilay, R., and Jaakkola, T. (2017). Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems*, pages 6833–6844.

Siddharthan, A. (2002). An architecture for a text simplification system. In *Language Engineering Conference, 2002. Proceedings*, pages 64–71. IEEE.

Sievert, C. and Shirley, K. (2014). Ldavis: A method for visualizing and interpreting topics. In *Proceedings of the workshop on interactive language learning, visualization, and interfaces*, pages 63–70.

So, D. R., Liang, C., and Le, Q. V. (2019). The evolved transformer. *arXiv preprint arXiv:1901.11117*.

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642. Citeseer.

Stajner, S., Franco-Salvador, M., Rosso, P., and Ponzetto, S. P. (2018). CATS: A Tool for Customized Alignment of Text Simplification Corpora. In chair), N. C. C., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Hasida, K., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S., and Tokunaga, T., editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Stajner, S. and Nisioi, S. (2018). A detailed evaluation of neural sequence-to-sequence models for in-domain and cross-domain text simplification. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*. European Language Resource Association.

Sulem, E., Abend, O., and Rappoport, A. (2018). BLEU is not suitable for the evaluation of text simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 738–744, Brussels, Belgium. Association for Computational Linguistics.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Suzuki, J. and Nagata, M. (2017). Cutting-off redundant repeating generations for neural abstractive summarization. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 291–297.

Tatalovic, M. (2018). AI writing bots are about to revolutionise science journalism: we must shape how this is done. *JCOM: Journal of Science Communication*, 17(1):C1–C1.

Teufel, S. and Moens, M. (2002). Summarizing scientific articles: experiments with relevance and rhetorical status. *Computational linguistics*, 28(4):409–445.

Tu, Z., Lu, Z., Liu, Y., Liu, X., and Li, H. (2016). Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, Berlin, Germany. Association for Computational Linguistics.

Uszkoreit, J., Ponte, J. M., Popat, A. C., and Dubiner, M. (2010). Large scale parallel document mining for machine translation. In *Proceedings of the*

23rd International Conference on Computational Linguistics, pages 1101–1109. Association for Computational Linguistics.

Vadapalli, R., Syed, B., Prabhu, N., Srinivasan, B. V., and Varma, V. (2018a). Sci-blogger: A step towards automated science journalism. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1787–1790. ACM.

Vadapalli, R., Syed, B., Prabhu, N., Srinivasan, B. V., and Varma, V. (2018b). When science journalism meets artificial intelligence: An interactive demonstration. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 163–168.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Wang, Q., Zhou, Z., Huang, L., Whitehead, S., Zhang, B., Ji, H., and Knight, K. (2018). Paper abstract writing through editing mechanism. *arXiv preprint arXiv:1805.06064*.

Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.

West, P., Holtzman, A., Buys, J., and Choi, Y. (2019). BottleSum: Unsupervised and self-supervised sentence summarization using the information bottleneck principle. 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 3750–3759, Hong Kong, China. Association for Computational Linguistics.

Wubben, S., Van Den Bosch, A., and Krahmer, E. (2012). Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 1015–1024. Association for Computational Linguistics.

Xie, Z., Genthial, G., Xie, S., Ng, A., and Jurafsky, D. (2018a). Noising and denoising natural language: Diverse backtranslation for grammar correction. 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 619–628, New Orleans, Louisiana. Association for Computational Linguistics.

Xie, Z., Genthial, G., Xie, S., Ng, A., and Jurafsky, D. (2018b). Noising and denoising natural language: Diverse backtranslation for grammar correction. 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 619–628. Association for Computational Linguistics.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. (2015a). Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057.

Xu, W., Callison-Burch, C., and Napoles, C. (2015b). Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

Xu, W., Napoles, C., Pavlick, E., Chen, Q., and Callison-Burch, C. (2016). Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Yang, Z., Hu, Z., Dyer, C., Xing, E. P., and Berg-Kirkpatrick, T. (2018). Unsupervised text style transfer using language models as discriminators. In *Advances in Neural Information Processing Systems*, pages 7287–7298.

Yih, W.-t., Goodman, J. T., Vanderwende, L. H., and Suzuki, H. (2010). Document summarization by maximizing informative content words. US Patent 7,702,680.

Zhang, J., Zhao, Y., Saleh, M., and Liu, P. J. (2019). Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *arXiv preprint arXiv:1912.08777*.

Zheng, H. and Lapata, M. (2019). Sentence centrality revisited for unsupervised summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6236–6247, Florence, Italy. Association for Computational Linguistics.

Zhu, J., Wang, Q., Wang, Y., Zhou, Y., Zhang, J., Wang, S., and Zong, C. (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.