2020

Separating Content Selection from Surface Realization in Neural Text Summarization

Logan Lebanoff
University of Central Florida

SEPARATING CONTENT SELECTION FROM SURFACE REALIZATION IN NEURAL TEXT SUMMARIZATION

by

LOGAN THIEN LEBANOFF
B.S. University of Central Florida, 2016
M.S. University of Central Florida, 2019

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2020

Major Professor: Fei Liu

ABSTRACT

Text summarization is a rapidly growing field with many new innovations. End-to-end models using the sequence-to-sequence architecture achieve high scores according to automatic metrics on standard datasets. However, they frequently generate summaries that are factually inconsistent with the original article – a vital problem to be solved before the summaries can be used in real-world applications. In addition, they are not generalizable to new domains, especially those with few training examples. In this dissertation, we propose to explicitly separate the two steps of content selection and surface realization in summarization. **Content selection** is the process of choosing important words/phrases/sentences from the document. **Surface realization** is the transformation of the selected content into a coherent, grammatical text summary. This paradigm more closely follows human patterns of summarization, as a human will often find important ideas within the article (content selection), and then write out a summary based on those ideas (surface realization). We make several contributions to the summarization field using this paradigm of separate content selection and surface realization steps. First, we present two techniques focusing on content selection: a model that can rank both single sentences and pairs of sentences in a unified space and a cascade approach that highlights salient words/phrases from sentences. Second, we present several studies on sentence fusion in summarization: an analysis of the quality of state-of-the-art summarizers for performing sentence fusion, a dataset containing points of correspondence between sentences, and a method utilizing these points of correspondence to improve sentence fusion. Finally, we introduce two methods with separate content selection and surface realization steps for multi-document summarization: a technique to adapt single document summarizers to the multi-document setting based on the Maximal Marginal Relevance (MMR) algorithm and a conceptual framework to model asynchronous endorsement between synopses and documents.

To my soon-to-be wife Tryphina,

who has always believed in me.

ACKNOWLEDGMENTS

I am incredibly thankful for the many people who have encouraged me in my PhD journey. I would like to thank my advisor Prof. Fei Liu. I am fortunate to have had an amazingly talented, knowledgeable, and encouraging advisor to guide me. She was always willing to put in the time and effort to help me succeed. Even when it meant getting down into the nitty-gritty, like checking my code, helping with writing papers, or giving me tips on networking and conference presentations. It is rare to have an advisor who will do that for her students. I truly admire her and have learned so much from her example.

I would like to thank my fiancée Tryphina, who has pushed me to keep going through my whole PhD. I wouldn't have made it to the end without her. Whenever I've felt inadequate in my field, or had bouts of imposter syndrome, she's reminded me of what I've accomplished. She cheered me on at every success. She and her family have always been supportive and have been so happy for me when good things happen, which has helped me more than they know.

I am very grateful for my twin brother Lance, who has been by my side throughout practically my entire life. Going through anything in life was easier and more fun with him there. It was encouraging to live with him and my roommates, Kevn and Tom, who were all in grad school, where we could laugh with each other and complain about similar experiences we were going through. It was invaluable to have friends to work hard with, and who were always down to play some volleyball or TowerFall whenever we needed a break.

I want to show my appreciation for my parents for instilling in me a hard work-ethic. They raised me to value education and learning, and that really translated to where I am now. My father always followed my interests and encouraged me to grow in the things I enjoyed. He saw that my brother and I liked computer games so much that he got us programs for us to create our own

games, and that's led me now to my PhD in computer science. I also want to thank my four older sisters, who I always looked up to. I've seen their success in school and sought to emulate that as I was growing up.

My mother has especially helped me strengthen my faith in God, for which I will forever be thankful to her. My Savior has given me all the talents, opportunities, and successes throughout my life, and there's no way I can ever repay Him for the wonderful things He's done for me.

I would also like to thank my committee members: Prof. Ulas Bagci, Prof. Niels Lobo, and Prof. Nazanin Rahnavard. Lastly, I would like to express my gratitude to those who have contributed to funding me during my PhD, including the National Science Foundation (grant IIS-1909603), and UCF for the Presidential Fellowship.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

The amount of information in the form of text stored online has been growing since the birth of the Internet and continues to increase. With so much knowledge available at our fingertips, the only bottleneck to internalizing that knowledge is the time spent digesting that information. Especially in the current era of social media, people expect quick bites of information. The field of automatic text summarization seeks to tackle this by slimming large texts down into more manageable summaries.

Text summarization has endless real-word uses. News articles can be reduced to a few sentences describing only the most salient or interesting highlights. Financial market analysts can make quicker investment decisions by having relevant news automatically summarized and digested. An online product may have thousands of reviews which can be consolidated into a single, informative meta-review. The salient ideas of academic papers can be extracted, which can be especially important for fast-moving areas of research, such as artificial intelligence and medicine. A summarization system can generate minutes for meetings automatically, relieving the need for a participant to take notes. Hundreds or thousands of free-text responses from surveys can be summarized into the main themes that are present. These examples and countless others demonstrate the potential benefit that automatic summarization can provide to humanity.

1.1 Extractive vs Abstractive Summarization

There exist two overarching approaches for text summarization. Extractive summarization directly copies content straight from the source document and places it in the summary. The model may copy whole sentences or copy words/phrases. One can think of it as using a highlighter to point out the important parts of the document. This is the approach employed by many classical summarization works, especially before the advent of neural networks [2, 3, 4, 5, 6].

Abstractive summarization creates a summary without being limited to only words and sentences within the source document. Often this is done by generating a summary one word at a time by picking a word from a set vocabulary, until a whole summary has been created. One can think of it as how a human might write a summary in their own words. This method allows for more compression since lengthy sentences can be reworded into simpler expressions. Most current summarization methods are abstractive [7, 8, 9, 10, 11, 12, 13, 14, 15].

Summarization techniques have improved greatly in recent years, especially with the advent of techniques such as copy mechanisms [11], multi-headed attention [16, 17], and pre-trained language models [13, 14, 15]. Most of these models are trained in an end-to-end fashion. End-to-end abstractive summarization models must perform two tasks implicitly at the same time. *Content selection:* important sentences or words/phrases from the source text must be selected [18, 19]. *Surface realization:* a summary must be generated which successfully merges the selected content together.

Because end-to-end models have this large amount of responsibility, they often generate two types of problematic outputs. First, they take the simplest route – copying whole sentences or large chunks of text without much change [11]. Essentially these models learn to only perform sentence selection and coarse content selection, while avoiding any complex rearranging or fusion of content. Second, when they do attempt to perform complex abstraction, they do so incorrectly [20, 21]. We hypothesize that these issues arise from expecting an abstractor to perform too many tasks at once, which prevents the model from learning to generate and merge content more intricately.

In this dissertation, we propose a more controllable approach by dividing the tasks with another model. Other works have also attempted similar decoupling of responsibilities from summarization models [22, 18]. This often takes the form of a pipelined approach with two separate models – a content selection model and a surface realization model. The content selection model chooses words, phrases, or full sentences that are deemed important. Next, the surface realization model merges the content together to create an abstractive summary. In the following section, we provide several benefits to this separation of tasks in text summarization.

1.2 Benefits of Separating of Content Selection from Surface Realization

In this dissertation, we seek to improve abstractive summarization by separating it into its two component tasks: content selection and surface realization. Content selection is the process of determining which text spans from the document are both important and non-redundant. This is often done at the sentence level, by extracting several sentences from the document, but it can also be done at a more fine-grained level, by selecting words or phrases from the document. Surface realization is the creation of the summary based on the information that was extracted during the content selection stage. It is often performed using a language model architecture by generating the output summary one word at a time.

An important question to answer is: what is the benefit of separating content selection from surface realization? Most summarization systems proposed recently have an end-to-end architecture, meaning they use one network to handle the entire process of summarization. They are given the source document as input and produce a summary as output. In the following, we present several benefits to separating content selection from surface realization in summarization.

Generalization First, the splitting up of tasks is more generalizable. In neural end-to-end summarization, models must be trained on thousands or millions of document-summary pairs to obtain reasonable performance. These models are often trained on news corpora, due to the abundance of

such document-summary pairs. However, to be able to summarize a different domain – say, Twitter posts – an entirely new model must be trained on Twitter post-summary pairs. These end-to-end models are not generalizable to new domains. In addition, many domains suffer from a scarcity of data, making training neural end-to-end models difficult. For example, meeting summarization – drawing out the main points of what was discussed in a meeting between several members – has only very small annotated datasets of tens or hundreds of examples [23, 24]. The separation of content selection and surface realization can alleviate these problems. A content selector can be trained to extract important sentences from documents. This content selector may not require a large number of document-summary pairs to be effectively trained. A surface realization model can be trained to compress single sentences or fuse multiple sentences together. This can be trained on a different domain that has a greater availability of data, such as news. The reason this can be done is because surface realization is generally not very dependent on the domain – sentences are often compressed or fused in similar ways whether it comes from the news or Twitter domains. In Chapter 5, we demonstrate how an abstractive summarizer trained on single-document summarization can be adapted to perform surface realization in the multi-document setting.

Follows Human Patterns Second, the separation of tasks seems to follow human patterns of summarization. It can be imagined that when a person is asked to summarize a news article, they will first read through the article. Then they will look back through the article and find one or two important sentences from the article, and then write out a summary sentence based on those sentences. This paradigm coincides with the steps of content selection (finding important sentences) and surface realization (writing out a summary sentence). These two steps are repeated until the important information has been covered. In addition, our analysis of several news datasets shows that humans do seem to follow this pattern. In Chapter 3, we find that 60-85% of summary sentences created by humans are created by fusing one or two sentences from the source document.

Factual Consistency Finally, separating content selection from surface realization has the potential to improve summaries' factual consistency. Factual consistency is a measure of how often the generated summaries introduce new meanings not present in the source document. In other words, it is whether the summary is true to the original document. Factual consistency is vital in order for automatically generated summaries to be applied to real-word problems. Even a single factual error can lead to huge consequences, especially in more sensitive fields, such as medical or military applications. End-to-end models have been shown to generate false facts frequently. Falke et al. [21] and Kryscinski et al. [25] analyzed the generated summaries of several state-of-the-art abstractive summarization systems, finding that 25% and 30%, respectively, of the summaries were inconsistent with the original document. We find that this problem is further exacerbated when models attempt to fuse multiple sentences together. In Chapter 4, we show that 38.3% of summary sentences that are generated from multiple sentences result in inconsistencies with the original [26]. This is a huge problem that is actively being researched in the community. Our approach is to separate content selection from surface realization. A content selection model can more accurately select multiple sentences from the article that are compatible with each other. A surface realization model can be trained specifically on faithfully fusing sentences together rather than overloading it with performing content selection at the same time.

1.3 Contribution of the Proposed Work

To explore methods of separating content selection from surface realization, we make use of several classical machine learning and novel neural methods. We explore these strategies on a variety of datasets, including single-document and multi-document datasets. The contribution of this thesis are:

- While most previous summarization works focus on either selecting single sentences from the source text or multiple sentences from the source text to fuse together, we present a model that can rank both single sentences and pairs of sentences in a unified space. In an analysis

of three summarization datasets, we find that a 60-85% of the time, humans select a sentence singleton or pair from the source document (content selection) and then compress or fuse the sentences together (surface realization). We attempt to model this behavior by fine-tuning BERT to give a score of how summary-worthy each sentence singleton/pair is. Then the highest scoring singletons/pairs are given to an abstractive summarizer to fuse the sentences together to form a summary sentence. Experiments show promise for this scoring method.

- We present a cascaded approach to content selection and surface realization. For content selection, we not only select important sentences from the source, but also tag individual words and phrases deemed salient from those sentences. The whole sentences are fed to an abstractive model, along with the tags informing which words/phrases to focus on. This cascaded sentence-level + word-level approach leads to significant gains compared to baselines that only perform content selection on the sentence-level.
- We analyze the quality of sentence fusions produced by five state-of-the-art abstractive summarization models. Results show that 38.3% of the sentence fusions are factually incorrect and 21.6% are grammatically incorrect. This demonstrates the importance of creating models that can effectively select mergeable content, and can fuse the content together properly.
- We introduce a new dataset of sentence fusion examples containing what we call *points of correspondence* between sentences. Points of correspondence are segments of text that represent what ties two sentences together. Our data can be useful to future research as a testbed for sentence fusion models, and the points of correspondence data can be analyzed to better understand how humans easily perform sentence fusion.
- We present methods for fusing sentences together using points of correspondence. We show that two approaches can be used to enhance Transformer model architectures leading to improved summary quality.
- We present a separate content selection method for multi-document summarization that can be applied to pre-trained single-document abstractive summarizers. The method is based on the Maximal Marginal Relevance (MMR) algorithm, and is used to select several sentences from

the source documents that are both salient and non-redundant. Based on which sentences are selected, a neural abstractive summarization model can be altered to attend only to the selected sentences. This retains the surface realization component of the abstractive model, while the content selection is performed by a separate module – MMR. This method can be used to transfer trained single-document summarization models to the multi-document setting without any additional training. Our experiments show a large improvement over previous abstractive models and most extractive methods on multi-document datasets.

- We introduce a conceptual framework that leverages the endorsement effect for multi-document summarization, which is described as follows. When an idea is repeated in multiple documents in the same cluster, it is likely that this idea is salient and should be included in the summary. Thus, document *A* endorses ideas present in other documents if document *A* also contains that idea. Our framework models this asynchronous endorsement between documents. Experiments on three multi-document summarization datasets show the efficacy of our framework.

CHAPTER 2: RELATED WORK

2.1 Extractive Summarization

Classical methods for text summarization have been extractive. Important sentences are extracted from a set of source documents and optionally compressed to form a summary [2, 3, 4, 5, 6, 27, 28, 29, 30, 31, 32]. In recent years neural networks have been exploited to learn word/sentence representations for single- and multi-document summarization [33, 34, 35, 36, 37]. These approaches remain extractive; and despite encouraging results, summarizing a large quantity of texts still requires sophisticated abstraction capabilities such as generalization, paraphrasing and sentence fusion.

2.2 Abstractive Summarization

Neural abstractive summarization utilizing the encoder-decoder architecture has shown promising results but studies focus primarily on single-document summarization [8, 38, 39, 40, 9, 41, 42, 10, 11, 12, 43]. The pointing mechanism [44, 45, 11] allows a summarization system to both copy words from the source text and generate new words from the vocabulary. Reinforcement learning is exploited to directly optimize evaluation metrics [10, 22, 46]. These studies focus on summarizing single documents in part because the training data are abundant.

Recently with the introduction of BERT [47], large pre-trained models have achieved state-of-the-art results on standard summarization datasets. Liu and Lapata [17] use BERT as an extractive

model to obtain representation of sentences in the document. They also experiment with an added Transformer decoder to produce abstractive summaries. Pretrained language models are shown to be able to perform relatively well in the zero-shot setting [48]. Currently, the best performing models use the encoder-decoder architecture with pre-training [13, 14, 15]. The models are trained to reconstruct a corrupted text, which aligns more closely to the summarization task than the standard language modelling or masked language modelling objectives.

2.3 Content Selection

Content selection is integral to any summarization system. Neural approaches to abstractive summarization often perform content selection jointly with surface realization using an encoder-decoder architecture, as described in the previous section. Training these models end-to-end means learning to perform both tasks simultaneously and can require a massive amount of data that is unavailable and unaffordable for many summarization tasks.

Recent approaches emphasize the importance of separating content selection from summary generation for abstractive summarization. Studies exploit extractive methods to identify content words and sentences that should be part of the summary and use them to guide the generation of abstracts [9, 18, 46, 22, 49, 50]. On the other hand, surface lexical features have been shown to be effective in identifying pertinent content [51, 52, 53]. Examples include sentence length, position, centrality, word frequency, whether a sentence contains topic words, and others. The surface cues can also be customized for new domains relatively easily. In Section 3.1, we present a step forward in this direction, where we focus on developing lightweight models to select summary-worthy sentence singletons and pairs and use them as the basis for summary generation.

A succinct sentence can be generated by shortening or rewriting a lengthy source text. Recent studies have leveraged neural encoder-decoder models to rewrite the first sentence of an article to a title-like summary [8, 42, 54, 55, 56, 57]. Compressive summaries can be generated in a similar vein by selecting important source sentences and then dropping inessential sentence elements such

as prepositional phrases. Before the era of deep neural networks it has been an active area of research, where sentence selection and compression can be accomplished using a pipeline or a joint model [2, 3, 4, 29, 27, 58, 31]. A majority of these studies focus on selecting and compressing sentence *singletons* only.

In Section 3.1, our approach teaches the system to determine if a sentence singleton or a pair should be selected to produce a summary sentence. A sentence pair (A, B) is preferred over its consisting sentences if they carry complementary content. Sentence B contains a reference ("*the attack*") and A contains a more complete description for it ("*bombing that killed 58*"). Sentences A and B each contain certain valuable information, and an appropriate way to merge them exists. As a result, a sentence pair can be scored higher than a singleton given the content it carries and compatibility of its consisting sentences.

There is a variety of successful summarization applications but few can afford to have a large number of annotated examples that are sufficient to meet the requirement of end-to-end neural abstractive summarization. Examples range from summarizing radiology reports [59, 60] to congressional bills [61] and meeting conversations [62, 63, 64]. The lack of annotated resources suggests that end-to-end systems may not be a "one-size-fits-all" solution to neural text summarization. There is an increasing need to develop cascaded architectures to allow for customized content selectors to be combined with general-purpose neural text generators to realize the full potential of neural abstractive summarization. We advocate for explicit content selection as it allows for a rigorous evaluation and visualization of intermediate results of such a module, rather than associating it with text generation. However, content selection concerns not only the selection of important segments from a document, but also the cohesiveness of selected segments and the amount of text to be selected in order for a neural text generator to produce a summary.

In Section 3.2, we aim to investigate the feasibility of a cascade approach to neural text summarization. We explore a constrained summarization task, where an abstract is created one sentence at a time through a cascaded pipeline. Our pipeline architecture chooses one or two sentences from the source document, then highlights their summary-worthy segments and uses those as a basis for composing a summary sentence. When a pair of sentences are selected, it is important to ensure

that they are *fusible*—there exists cohesive devices that tie the two sentences together into a coherent text—to avoid generating nonsensical outputs [65, 66]. Highlighting sentence segments allows us to perform fine-grained content selection that guides the neural text generator to stitch selected segments into a coherent sentence.

2.4 Surface Realization and Sentence Fusion

Prior to deep learning, abstractive summarization and surface realization has been investigated [67, 68, 69, 70, 71, 72, 73, 74, 75]. These approaches construct domain templates using a text planner or an open-IE system and employ a natural language generator for surface realization. Limited by the availability of labelled data, experiments are often performed on small domain-specific datasets.

Sentence fusion aims to produce a single summary sentence by fusing multiple source sentences. However, many aspects of this approach are largely underinvestigated, such as determining the set of source sentences to be fused, handling its large cardinality, and identifying the sentence relationships for performing fusion. Dependency graphs and discourse structure have proven useful for aligning and combining multiple sentences into a single sentence [76, 77, 78, 79, 70]. Previous studies assume a set of similar source sentences can be gathered by clustering sentences or by comparing to a reference summary sentence [80, 81, 82, 83, 75]; but these methods can be suboptimal. Joint models for sentence selection and fusion implicitly perform content planning [84, 6, 73, 32] and there is limited control over which sentences are merged and how. Mehdad et al. [62] construct an entailment graph over sentences for sentence selection, then fuse sentences together using a word graph. Abstract meaning representation and other graph-based representations have also shown success in sentence fusion [74, 19]. Geva et al. [65] fuse pairs of sentences together using Transformer, focusing on discourse connectives between sentences.

Recent summarization research has put special emphasis on faithfulness to the original text. Cao et al. [20] use seq-to-seq models to rewrite templates that are prone to including irrelevant entities. Incorporating additional information into a seq-to-seq model, such as entailment and de-

pendency structure, has proven successful [85, 55]. The closest work to our human evaluation seems to be from Falke et al. [21]. Similar to our work, they find that the PG model is more faithful than Fast-Abs-RL and Bottom-Up, even though it has lower ROUGE. They show that 25% of outputs from these state-of-the-art summarization models are unfaithful to the original article. Cao et al. [20] reveal a similar finding that 27% of the summaries generated by a neural sequence-to-sequence model have errors. Kryscinski et al. [25] find that 30% of summary outputs contained factual inconsistencies with the original article. In Section 4.1, we perform an analysis of five state-of-the-art summarizers on their ability to perform sentence fusion accurately. In contrast to other studies, we limit our study to only summary sentences created by *fusion*. We find 38% of sentence fusions made by automatic summarizers to be unfaithful. Our work examines a wide variety of state-of-the-art summarization systems, and perform in-depth analysis over other measures including grammaticality, coverage, and method of merging.

Uncovering hidden correspondences between sentences is essential for producing proper summary sentences. A number of recent efforts select important words and sentences from a given document, then let the summarizer attend to selected content to generate a summary [18, 49, 46, 86, 50], as described in Section 2.3. These systems are largely agnostic to sentence correspondences, which can have two undesirable consequences. If only a single sentence is selected, it can be impossible for the summarizer to produce a fusion sentence from it. Moreover, if *non-fusible* textual units are selected, the summarizer is forced to fuse them into a summary sentence, yielding output summaries that often fail to keep the original meaning intact. In Section 4.2, we investigate the correspondences between sentences to gain an understanding of sentence fusion.

Establishing correspondence between sentences goes beyond finding common words. Humans can fuse sentences sharing *few or no* common words if they can find other types of correspondence. Fusing such disparate sentences poses a serious challenge for automated fusion systems [77, 78, 87, 88, 89, 62, 19]. These systems rely on common words to derive a connected graph from input sentences or subject-verb-object triples [90]. When there are no common words in sentences, systems tend to break apart.

There has been a lack of annotated datasets and guidelines for sentence fusion. Few studies have investigated the types of correspondence between sentences such as entity and event coreference. Evaluating sentence fusion systems requires not only novel metrics [91, 92, 93, 94] but also high-quality ground-truth annotations. It is therefore necessary to conduct a first study to look into cues humans use to establish correspondence between disparate sentences.

We envision sentence correspondence to be related to text *cohesion* and *coherence*, which help establish correspondences between two pieces of text. Halliday and Hasan [95] describe text **cohesion** as cohesive devices that tie two textual elements together. They identify five categories of cohesion: *reference*, *lexical cohesion*, *ellipsis*, *substitution* and *conjunction*. In contrast, **coherence** is defined in terms of discourse relations between textual elements, such as *elaboration*, *cause* or *explanation*. Previous work studied discourse relations [65]. McKeown et al. [87] compile a corpus of 300 sentence fusions as a first step toward a supervised fusion system. However, the input sentences have very similar meaning, though they often present lexical variations and different details. A large-scale dataset of sentence fusions has been recently collected [65], where each sentence has disparate content and are connected by various discourse connectives.

In Section 4.2, we introduce a dataset of sentence fusion instances annotated with points of correspondence. It focuses on *text cohesion*, which plays a crucial role in generating proper fusion sentences. Our dataset contains pairs of source and fusion sentences collected from news editors in a natural environment. The work is particularly meaningful to text-to-text and data-to-text generation [96] that demand robust modules to merge disparate content.

A renewed emphasis must be placed on sentence fusion in the context of neural abstractive summarization. A majority of the systems are trained end-to-end [11, 10, 43, 46, 18, 50], as described in Section 2.2, where an abstractive summarizer is rewarded for generating summaries that contain the same words as human abstracts, measured by automatic metrics such as ROUGE [97]. A summarizer, however, is not rewarded for correctly fusing sentences. In fact, when examined more closely, only few sentences in system abstracts are generated by fusion [21]. For instance, 6% of summary sentences generated by Pointer-Gen [11] are through fusion, whereas human abstracts contain 32% fusion sentences. Moreover, sentences generated by fusion are prone to errors.

They can be ungrammatical, nonsensical, or otherwise ill-formed. There is thus an urgent need to develop neural abstractive summarizers to fuse sentences properly.

The importance of sentence fusion has long been recognized by the community before the era of neural text summarization. The pioneering work of Barzilay et al. [67] introduces an information fusion algorithm that combines similar elements across related text to generate a succinct summary. Later work, such as [77, 78, 88, 28, 62], builds a dependency or word graph by combining syntactic trees of similar sentences, then employs integer linear programming to decode a summary sentence from the graph. Most of these studies have assumed a set of *similar* sentences as input, where fusion is necessary to reduce repetition. Nonetheless, humans do not limit themselves to combine similar sentences. In Section 4.3, we pay particular attention to fuse *disparate* sentences that contain fundamentally different content but remain related to make fusion sensible [88].

2.5 Multi-Document Summarization

In recent years, multi-document abstractive summarization (MuDAS) has been a large focus in the summarization field. Many multi-document datasets contain few training examples, however. One approach to tackle this issue is to train an encoder-decoder model on data-rich single-document datasets, then transfer the model to a multi-document setting. In particular, Baumel et al. [98] propose to extend an abstractive summarization system to generate query-focused summaries; Zhang et al. [99] add a document set encoder to their hierarchical summarization framework. With these few exceptions, little research has been dedicated to investigate the feasibility of extending the encoder-decoder framework to generate abstractive summaries from multi-document inputs, where available training data are scarce.

In Section 5.1, we present some first steps towards the goal of extending the encoder-decoder model to a multi-document setting. We introduce an adaptation method combining the pointer-generator (PG) networks [11] and the maximal marginal relevance (MMR) algorithm [100]. The PG model, trained on SDS data and detailed in Section §5.1.2, is capable of generating document

abstracts by performing text abstraction and sentence fusion. However, if the model is applied at test time to summarize multi-document inputs, there will be limitations. Our PG-MMR algorithm teaches the PG model to effectively recognize important content from the input documents, hence improving the quality of abstractive summaries, all without requiring any training on multi-document inputs.

Saliency is one of the most important characteristics of summaries. In MuDAS, saliency of a text segment is measured by its frequency, which prefers frequent occurrence in a set of documents [101]. Optimizing summaries for saliency and non-redundancy using probabilisttic graphical models [102], integer linear programming [6] and determinantal point processes [103] have attained considerable success prior to deep neural models. These optimization methods can effectively model frequency, but produce extractive rather than abstractive summaries from multi-document inputs.

It remains to be seen whether deep neural models can adequately represent frequency, particularly of named entities and quantities that carry little semantic meaning. Frequency is rarely modeled in single-document summarization [11, 9, 37, 104, 14, 105], in part because single documents are concise and contain little redundancy. In recent MuDAS studies, Liu and Lapata [50] encode source documents using hierarchical Transformers where cross-document relationships are captured by attention. Perez-Beltrachini et al. [106] explore structured convolutional decoders. Li et al. [107] leverage similarity and discourse graphs to alter the attention mechanism of neural encoder-decoder models. Bražinskas et al. [108] use few-shot learning to bootstrap summary generation. However, without explicitly modeling frequency, existing neural methods remain incapable of differentiating entities or quantities and they may fail at accurately recognizing salient details for MuDAS.

In Section 5.2, we are particularly interested in condensing multiple source documents into a single document, then consolidate the content into an abstract [19, 109]. We enhance the single document with fine-grained segment salience to offset the lead bias [18, 110], which hinders the development of multiple-document summarization. Our salience estimates are obtained from a frequency-driven endorsement model. Frequency and redundancy are essential in multi-document

summarization. Without these, even humans tend to disagree on what information is relevant and should be retained in the summary [111].

2.6 Datasets

Early summarization research was evaluated on the Document Understanding Conference (DUC) and Text Analysis Conference (TAC) datasets [112, 113]. A new data iteration was created each year between 2001-2011. These datasets are for multi-document summarization and are annotated by humans, containing only hundreds of training pairs. (see Table 2.1 for a comparison to standard single-document summarization datasets). The summarization system is tasked with generating a concise, fluent summary of 100 words or less from a set of 10 documents discussing a topic. Each system summary is compared against 4 human abstracts created by NIST assessors. The cost to create ground-truth summaries from multiple-document inputs can be prohibitive. The MDS datasets are thus too small to be used to train neural encoder-decoder models with millions of parameters without overfitting. This makes these datasets incredibly challenging, but they are also the highest quality because they were created by experts.

More recent datasets are collected automatically by scraping websites for documents and summaries, allowing for datasets containing hundreds of thousands or millions of examples. Most of these datasets are only for the news domain, only are for the single-document setting, and contain only one ground-truth summary. The most commonly used dataset is CNN/Daily Mail [114]. The training set contains about 287k article-summary pairs and the test set contains 11k pairs. The task is to reduce a news article to a multi-sentence summary (4 sentences on average).

Another common dataset is XSum [43], a dataset created for extreme, abstractive summarization. The task is to reduce a news article to a short, one-sentence summary. Both source articles and reference summaries are gathered from the BBC website. The training set contains about 204k article-summary pairs and the test contains 11k pairs.

Table 2.1: A comparison of datasets

DATASET	SOURCE	SUMMARY	#PAIRS
Gigaword [7]	**the first sentence** of a news article	**8.3 words** title-like	**4 Million**
CNN/Daily Mail [114]	**a news article**	**56 words** multi-sent	**312 K**
TAC (08-11) (Dang et al., 2008)	**10 news articles** related to a topic	**100 words** multi-sent	**728**
DUC (03-04) [112]	**10 news articles** related to a topic	**100 words** multi-sent	**320**

A comparison of datasets available for sent. summarization (Gigaword), single-doc (CNN/DM) and multi-doc summarization (DUC/TAC). The labelled data for multi-doc summarization are much less.

Other single-document datasets include Gigaword (news headlines) [7], Newsroom (news) [115], arXiv (scientific papers) [116], PubMed (scientific papers) [116], BIGPATENT (patents) [117], WikiHow (how-to articles) [118], Reddit TIFU (social media stories) [119], AESLC (emails) [120], and BillSum (legislation) [61].

A few other multi-document datasets have been introduced as well. WikiSum [121] contains over 2 million examples, where each example summary is a Wikipedia article and the input documents are the reference links on the webpage. Similarly, Multi-News [122] contains over 50k examples, where each example is a summary from newser.com and the input documents are news article links contained in the summary web page. The Wikipedia Current Events Portal (WCEP) dataset [123] contains 10k examples taken from a current events page on Wikipedia. A one-sentence summary about a news event is written by experienced writers and 1-2 articles are linked to it. The input documents are augmented with similar articles drawn from Common Crawl[1].

[1] https://commoncrawl.org/2016/10/news-dataset-available/

2.7 Metrics

Evaluation of summarization models is a huge challenge. Numerous different summaries may be valid summaries for a given document. A summary can be worded differently and still have the same meaning. Separate individuals may have differing views on what parts of a document are important [25]. A summary should be brief but still contain the important points of a document. It should not modify or introduce new information that was not contained in the original document. All of these dimensions should be considered when evaluating summaries, and it can be difficult even for humans be consistent judges of quality [25].

The standard evaluation metric, ROUGE [97], measures the word overlap between the system summary and one or more reference summaries. Several variants exist including the overlap of unigrams (ROUGE-1), bigrams (ROUGE-2), skip bigrams with a maximum distance of 4 words (ROUGE-SU4), and longest common subsequence (ROUGE-L). While widely used in summarization, this metric is known to have issues. It has relatively weak correlation with human judgements, and it does not handle summaries that vary lexically but have equal semantic meaning. Several automatic metrics have been proposed to better handle synonyms [124, 125, 126, 127, 128], but these metrics still struggle with highly abstractive summaries. Automatic metrics for evaluating the factual consistency of summaries using question-answering models have been proposed [129, 94, 93].

CHAPTER 3: CONTENT SELECTION

The primary purpose of summarization is to present the important content in a condensed form and leave out auxiliary details. Thus, it is necessary for automatic summarization models to be able to select the appropriate content from the document. While most current methods do some form of content selection implicitly using an end-to-end model, we show that incorporating an explicit content selection step can improve summarization performance. In this chapter, we present two content selection techniques.

In Section 3.1, we propose a framework for scoring sentence singletons and pairs in the same space for content selection. Previous work selected either a single sentence only or multiple sentences only. Humans, however, will usually choose 1 or 2 sentences from an article to then fuse together to form a summary sentence. We show that our content selection model using BERT outperforms previous methods on selecting the salient sentences.

In Section 3.2, we propose a cascaded architecture for content selection. We use a model that will select a sentence singleton or pair from the document, while simultaneously selecting words/phrases from the selected sentence(s). This architecture results in improvement in ROUGE scores over models that select sentences only.

3.1 Selecting Sentence Singletons and Pairs for Abstractive Summarization

When writing a summary, humans tend to choose content from one or two sentences and merge them into a single summary sentence. However, the mechanisms behind the selection of *one*

or *multiple* source sentences remain poorly understood. Sentence fusion assumes multi-sentence input; yet sentence selection methods only work with single sentences and not combinations of them. There is thus a crucial gap between sentence selection and fusion to support summarizing by both compressing single sentences and fusing pairs. This section attempts to bridge the gap by ranking sentence singletons and pairs together in a unified space. Our proposed framework attempts to model human methodology by selecting either a single sentence or a pair of sentences, then compressing or fusing the sentence(s) to produce a summary sentence. We conduct extensive experiments on both single- and multi-document summarization datasets and report findings on sentence selection and abstraction.[1]

3.1.1 Introduction

Abstractive summarization aims at presenting the main points of an article in a succinct and coherent manner. To achieve this goal, a proficient editor can rewrite a source sentence into a more succinct form by dropping inessential sentence elements such as prepositional phrases and adjectives. She can also choose to fuse multiple source sentences into one by reorganizing the points in a coherent manner. In fact, it appears to be common practice to summarize by either compressing single sentences or fusing multiple sentences. We investigate this hypothesis by analyzing human-written abstracts contained in three large datasets: DUC-04 [112], CNN/Daily Mail [114], and XSum [43]. For every summary sentence, we find its *ground-truth set* containing one or more source sentences that exhibit a high degree of similarity with the summary sentence (details in §3.1.3). As shown in Figure 3.1, across the three datasets, 60-85% of summary sentences are generated by fusing one or two source sentences.

Selecting summary-worthy sentences has been studied in the literature, but there lacks a mechanism to weigh sentence singletons and pairs in a unified space. Extractive methods focus on

[1]This section is adapted from: L. Lebanoff, K. Song, F. Dernoncourt, D. S. Kim, S. Kim, W. Chang, and F. Liu, Scoring Sentence Singletons and Pairs for Abstractive Summarization, in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL), 2019.

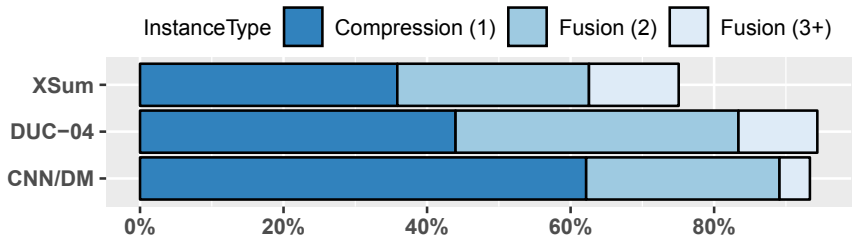

Figure 3.1: Portions of summary sentences generated by compression (content is drawn from 1 source sentence) and fusion (content is drawn from 2 or more source sentences). Humans often grab content from 1 or 2 document sentences when writing a summary sentence.

selecting sentence singletons using greedy [100], optimization-based [4, 130, 131], and non-autoregressive methods [33, 132]. In contrast, existing sentence fusion studies tend to assume ground sets of source sentences are already provided, and the system fuses each set of sentences into a single one [111, 81, 89]. There is thus a crucial gap between sentence selection and fusion to support summarizing by both compressing single sentences and fusing pairs. This section attempts to bridge the gap by ranking singletons and pairs together by their likelihoods of producing summary sentences.

The selection of sentence singletons and pairs can bring benefit to neural abstractive summarization, as a number of studies seek to separate content selection from summary generation [46, 49, 18, 133]. Content selection draws on domain knowledge to identify relevant content, while summary generation weaves together selected source and vocabulary words to form a coherent summary. Despite having local coherence, system summaries can sometimes contain erroneous details [11] and forged content [20, 55]. Separating the two tasks of content selection and summary generation allows us to closely examine the compressing and fusing mechanisms of an abstractive summarizer.

In this section we propose a method to learn to select sentence singletons and pairs, which then serve as the basis for an abstractive summarizer to compose a summary sentence-by-sentence, where singletons are shortened (i.e., compressed) and pairs are merged (i.e., fused). A sentence pair (A, B) is preferred over its consisting sentences if they carry complementary content. Table 3.1 shows an example. Sentence B contains a reference ("*the attack*") and A contains a more complete

Table 3.1: Example sentence singleton and pair, before and after compression/merging.

Sentence Pair: (A) The bombing killed 58 people. (B) Wajid Shamsul Hasan, Pakistan's high commissioner to Britain, and Hamid Gul, former head of the ISI, firmly denied the agency's involvement in the attack.	**Merged Sentence:** Pakistan denies its spy agency helped plan bombing that killed 58.
Sentence Singleton: (A) Pakistani Maj. Gen. Athar Abbas said the report "unfounded and malicious" and an "effort to malign the ISI," – Pakistan's directorate of inter-services intelligence.	**Compressed Sentence:** Maj. Gen. Athar Abbas said the report was an "effort tomalign the ISI."

description for it ("*bombing that killed 58*"). Sentences A and B each contain certain valuable information, and an appropriate way to merge them exists. As a result, a sentence pair can be scored higher than a singleton given the content it carries and compatibility of its consisting sentences.

We exploit state-of-the-art neural representations and traditional vector space models to characterize singletons and pairs; we then provide suggestions on the types of representations useful for summarization. Experiments are performed on both single- and multi-document summarization datasets, where we demonstrate the efficacy of selecting sentence singletons and pairs as well as its utility to abstractive summarization. Our research contributions can be summarized as follows:

- the present study fills an important gap by selecting sentence singletons and pairs jointly, assuming a summary sentence can be created by either shortening a singleton or merging a pair. Compared to abstractive summarizers that perform content selection implicitly, our method is flexible and can be extended to multi-document summarization where training data is limited;
- we investigate the factors involved in representing sentence singletons and pairs. We perform extensive experiments and report findings on sentence selection and abstraction.[2]

3.1.2 Our Model

We present the first attempt to transform sentence singletons and pairs to real-valued vector representations capturing semantic salience so that they can be measured against each other (§3.1.2.1). This is a nontrivial task, as it requires a direct comparison of texts of varying length—a pair of

[2]We make our code and models publicly available at `https://github.com/ucfnlp/summarization-sing-pair-mix`

sentences is almost certainly longer than a single sentence. For sentence pairs, the representations are expected to further encode sentential semantic compatibility. In §3.1.2.2, we describe our method to utilize highest scoring singletons and pairs to a neural abstractive summarizer to generate summaries.

3.1.2.1 Scoring Sentence Singletons and Pairs

Given a document or set of documents, we create a set D of singletons and pairs by gathering all single sentences and arbitrary pairs of them. We refer to a singleton or pair in the set as an *instance*. The sentences in a pair are arranged in order of their appearance in the document or by date of documents. Let N be the number of single sentences in the input document(s), a complete set of singletons and pairs will contain $|D|=\frac{\mathsf{N}(\mathsf{N}-1)}{2}+\mathsf{N}$ instances. Our goal is to score each instance based on the amount of summary-worthy content it conveys. Despite their length difference, a singleton can be scored higher than a pair if it contains a significant amount of salient content. Conversely, a pair can outweigh a singleton if its component sentences are salient and compatible with each other.

Building effective representations for singletons and pairs is therefore of utmost importance. We attempt to build a vector representation for each instance. The representation should be invariant to the instance type, i.e., a singleton or pair. We exploit the BERT architecture [47] to learn instance representations. The representations are fine-tuned for a classification task predicting whether a given instance contains content used in human-written summary sentences (details for ground-truth creation in §3.1.3).

BERT BERT supports our goal of encoding singletons and pairs indiscriminately. It introduces two pretraining tasks to build deep contextual representations for words and sequences. A sequence can be a single sentence (A) or pair of sentences (A+B).[3] The first task predicts *missing*

[3]In the original BERT paper [47], a "sentence" is used in a general sense to denote an arbitrary span of contiguous text; we refer to an actual linguistic sentence.

words in the input sequence. The second task predicts if B is the *next sentence* following A. It requires the vector representation for (A+B) to capture the coherence of two sentences. As coherent sentences can often be fused together, we conjecture that the second task is particularly suited for our goal.

Concretely, BERT constructs an input sequence by prepending a singleton or pair with a "[CLS]" symbol and delimiting the two sentences of a pair with "[SEP]." The representation learned for the [CLS] symbol is used as an aggregate sequence representation for the later classification task. We show an example input sequence in Eq. (3.1). In the case of a singleton, w_i^{B} are padding tokens.

$$\{w_i\} = \texttt{[CLS]}, w_1^{\mathsf{A}}, w_2^{\mathsf{A}}, \ldots, \texttt{[SEP]}, w_1^{\mathsf{B}}, w_2^{\mathsf{B}}, \ldots, \texttt{[SEP]} \tag{3.1}$$

$$\mathbf{e}_i = \mathbf{e}_{\mathrm{w}}(w_i) + \mathbf{e}_{\mathrm{sgmt}}(w_i) + \mathbf{e}_{\mathrm{wpos}}(w_i) + \mathbf{e}_{\mathrm{spos}}(w_i) \tag{3.2}$$

In Eq. (3.2), each token w_i is characterized by an input embedding $\mathbf{e}_i$, calculated as the element-wise sum of the following embeddings:

- $\mathbf{e}_{\mathrm{w}}(w_i)$ is a *token* embedding;
- $\mathbf{e}_{\mathrm{sgmt}}(w_i)$ is a *segment* embedding, signifying whether w_i comes from sentence A or B.
- $\mathbf{e}_{\mathrm{wpos}}(w_i)$ is a *word position* embedding indicating the index of w_i in the input sequence;
- we introduce $\mathbf{e}_{\mathrm{spos}}(w_i)$ to be a *sentence position* embedding; if w_i is from sentence A (or B), $\mathbf{e}_{\mathrm{spos}}(w_i)$ is the embedding indicating the index of sentence A (or B) in the original document.

Intuitively, these embeddings mean that, the extent to which a word contributes to the sequence (A+B) representation depends on these factors: (i) word salience, (ii) importance of sentences A and B, (iii) word position in the sequence, and, (iv) sentence position in the document. These factors coincide with heuristics used in summarization literature [101], where leading sentences of a document and the first few words of a sentence are more likely to be included in the summary.

The input embeddings are then fed to a multi-layer and multi-head attention architecture to build deep contextual representations for tokens. Each layer employs a Transformer block [134], which introduces a self-attention mechanism that allows each hidden state $\mathbf{h}_i^l$ to be compared with every other hidden state of the same layer $[\mathbf{h}_1^l, \mathbf{h}_2^l, \ldots, \mathbf{h}_{\mathsf{N}}^l]$ using a parallelizable, multi-head attention mechanism (Eq. (3.3-3.4)).

$$\mathbf{h}_i^1 = f_{\text{self-attn}}^1(\mathbf{e}_i, [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_{\mathsf{N}}]) \tag{3.3}$$

$$\mathbf{h}_i^{l+1} = f_{\text{self-attn}}^{l+1}(\mathbf{h}_i^l, [\mathbf{h}_1^l, \mathbf{h}_2^l, \ldots, \mathbf{h}_{\mathsf{N}}^l]) \tag{3.4}$$

The representation at final layer L for the [CLS] symbol is used as the sequence representation $\mathbf{h}_{[\mathsf{CLS}]}^{\mathsf{L}}$. The representations can be fine-tuned with an additional output layer to generate state-of-the-art results on a wide range of tasks including reading comprehension and natural language inference. We use the pretrained BERT base model and fine-tune it on our specific task of predicting if an instance (a singleton or pair) $p_{\text{inst}} = \sigma(\mathbf{w}^\top \mathbf{h}_{[\mathsf{CLS}]}^{\mathsf{L}})$ is an appropriate one, i.e., belonging to the ground-truth set of summary instances for a given document. At test time, the architecture indiscriminately encodes a mixed collection of sentence singletons/pairs. We then obtain a likelihood score for each instance. This framework is thus a first effort to build semantic representations for singletons and pairs capturing informativeness and semantic compatibility of two sentences.

VSM We are interested in contrasting BERT with the traditional vector space model [135] for representing singletons and pairs. BERT learns instance representations by attending to important content words, where the importance is signaled by word and position embeddings as well as pairwise word relationships. Nonetheless, it remains an open question whether BERT can successfully weave the meaning of *topically important words* into representations. A word "border" is topically important if the input document discusses border security. A topic word is likely to be repeatedly mentioned in the input document but less frequently elsewhere. Because sentences containing topical words are often deemed summary-worthy [136], it is desirable to represent sentence singletons and pairs based on the amount of topical content they convey.

VSM represents each sentence as a sparse vector. Each dimension of the vector corresponds to an n-gram weighted by its TF-IDF score. A high TF-IDF score suggests the n-gram is important to the topic of discussion. We further strengthen the sentence vector with position and centrality information, i.e., the sentence position in the document and the cosine similarity between the sentence and document vector. We obtain a document vector by averaging over its sentence vectors, and we similarly obtain a vector for a pair of sentences. We use VSM representations as a baseline to contrast its performance with distributed representations from BERT. To score singletons and pairs, we use the LambdaMART model[4] which has demonstrated success on related NLP tasks [137]; it also fits our requirements of ranking singletons and pairs indiscriminately.

3.1.2.2 Generating Summaries

We proceed by performing a preliminary investigation of summary generation from singletons and pairs; they are collectively referred to as *instances*. In the previous section, a set of summary instances is selected from a document. These instances are treated as "raw materials" for a summary; they are fed to a neural abstractive summarizer which processes them into summary sentences via fusion and compression. This strategy allows us to separately evaluate the contributions from instance selection and summary composition.

We employ the MMR principle [100] to select a set of highest scoring and non-redundant instances. The method adds an instance $\hat{P}$ to the summary S iteratively per Eq. (3.5) until a length threshold has been reached. Each instance is weighted by a linear combination of its importance score $I(P_k)$, obtained by BERT or VSM, and its redundancy score $R(P_k)$, computed as the cosine similarity between the instance and partial summary. λ is a balancing factor between importance and redundancy.[5] Essentially, MMR prevents the system from selecting instances that are too

[4] `https://sourceforge.net/p/lemur/wiki/RankLib/`

[5] We use a coefficient λ of 0.6.

similar to ones already selected.

$$\hat{P} = \underset{P_k \in D \setminus S}{\arg\max} \Big[\lambda I(P_k) - (1-\lambda) R(P_k) \Big] \tag{3.5}$$

Composing a summary from selected instances is a non-trivial task. As a preliminary investigation of summary composition, we make use of pointer-generator (PG) networks [11] to compress/fuse sentences into summary sentences. PG is a sequence-to-sequence model that has achieved state-of-the-art performance in abstractive summarization by having the ability to both copy tokens from the document or generate new tokens from the vocabulary. When trained on document-summary pairs, the model has been shown to remove unnecessary content from sentences and can merge multiple sentences together.

In this work, rather than training on document-summary pairs, we train PG exclusively on ground-truth instances. This removes most of the responsibility of content selection, and allows it to focus its efforts on merging the sentences. We use instances derived from human summaries (§3.1.3) to train the network, which includes a sentence singleton or pair along with the ground-truth compressed/merged sentence. At test time, the network receives an instance from BERT or VSM and outputs a summary sentence, then repeats this process to generate several sentences. In Figure 4.7 we present an illustration of the system architecture.

3.1.3 Data

Our method does not require a massive amount of annotated data. We thus report results on single- and multi-document summarization datasets.

We experiment with (i) XSum [43], a dataset created for extreme, abstractive summarization. The task is to reduce a news article to a short, one-sentence summary. Both source articles and reference summaries are gathered from the BBC website. (ii) CNN/DM [114], an abstractive summarization dataset frequently exploited by recent studies. The task is to reduce a news article

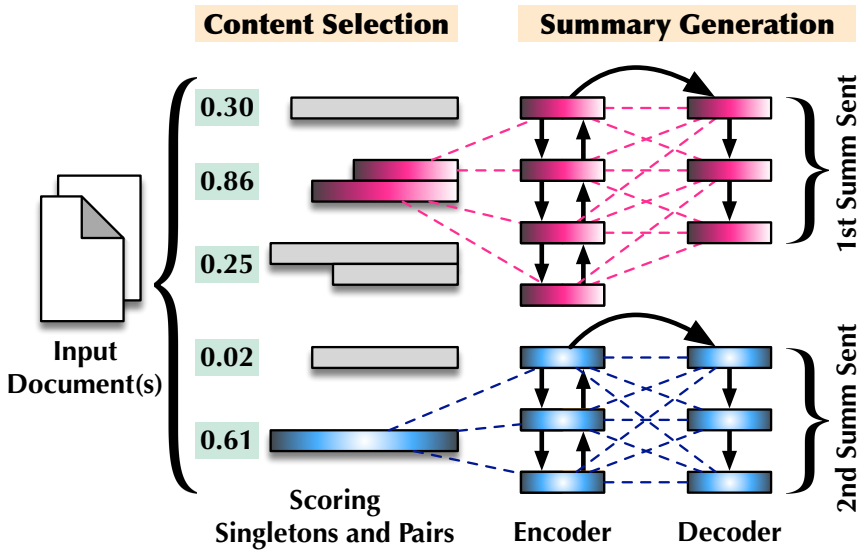

Figure 3.2: System architecture. In this example, a sentence pair is chosen (red) and then merged to generate the first summary sentence. Next, a sentence singleton is selected (blue) and compressed for the second summary sentence.

to a multi-sentence summary (4 sentences on average). We use the non-anonymzied version of the dataset. (iii) DUC-04 [112], a benchmark multi-document summarization dataset. The task is to create an abstractive summary (5 sentences on average) from a set of 10 documents discussing a given topic.

We build a training set for both tasks of content selection and summary generation. This is done by creating ground-truth sets of instances based on document-summary pairs. Each document and summary pair (D,S) is a collection of sentences $D = \{d_1, d_2, ..., d_M\}$ and $S = \{s_1, s_2, ..., s_N\}$. We wish to associate each summary sentence s_n with a subset of the document sentences $\tilde{D} \subseteq D$, which are the sentences that are merged to form s_n. Our method chooses multiple sentences that work together to capture the most overlap with summary sentence s_n, in the following way.

We use averaged ROUGE-1, -2, -L scores [97] to represent sentence similarity. The source sentence most similar to s_n is chosen, which we call $\tilde{d}_1$. All shared words are then removed from s_n to create s'_n, effectively removing all information already captured by $\tilde{d}_1$. A second source sentence $\tilde{d}_2$ is selected that is most similar to the remaining summary sentence s'_n, and shared words are again removed from s'_n to create s''_n. This process of sentence selection and overlap removal

is repeated until no remaining sentences have at least two overlapping content words (words that are non-stopwords or punctuation) with s_n. The result is referred to as a ground-truth set $(s_n, \tilde{D})$ where $\tilde{D} = \{\tilde{d}_1, \tilde{d}_2, ..., \tilde{d}_{|\tilde{D}|}\}$. To train the models, $\tilde{D}$ is limited to one or two sentences because it captures the large majority of cases. All empty ground-truth sets are removed, and only the first two sentences are chosen for all ground-truth sets with more than two sentences. A small number of summary sentences have empty ground-truth sets, corresponding to 2.85%, 9.87%, 5.61% of summary sentences in CNN/DM, XSum, and DUC-04 datasets. A detailed plot of the ground-truth set size is illustrated in Figure 3.1, and samples of the ground-truth are found in the supplementary.

We use the standard train/validation/test splits for both CNN/Daily Mail and XSum. We train our models on ground-truth sets of instances created from the training sets and tune hyperparameters using instances from the validation sets. DUC-04 is a test-only dataset, so we use the models trained on CNN/Daily Mail to evaluate DUC-04. Because the input is in the form of multiple documents, we select the first 20 sentences from each document and concatenate them together into a single mega-document [133]. For the sentence position feature, we keep the sentence positions from the original documents. This handling of sentence position, along with other features that are invariant to the input type, allows us to effectively train on single-document inputs and transfer to the multi-document setting.

3.1.4 Results

Evaluation Setup In this section we evaluate our proposed methods on identifying summary-worthy instances including singletons and pairs. We compare this scheme with traditional methods extracting only singletons, then introduce novel evaluation strategies to compare results. We exploit several strong extractive baselines: (i) *SumBasic* [138] extracts sentences by assuming words occurring frequently in a document have higher chances of being included in the summary; (ii) *KL-Sum* [102] greedily adds sentences to the summary to minimize KL divergence; (iii) *LexRank* [139] estimates sentence importance based on eigenvector centrality in a document graph representation.

Table 3.2: Instance selection results.

	System	Primary			Secondary			All		
		P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
CNN/Daily Mail	LEAD-Baseline	31.9	38.4	34.9	10.7	34.3	16.3	39.9	37.3	38.6
	SumBasic [138]	15.2	17.3	16.2	5.3	15.8	8.0	19.6	16.9	18.1
	KL-Summ (Haghighi et al., 2009)	15.7	17.9	16.7	5.4	15.9	8.0	20.0	17.4	18.6
	LexRank [139]	22.0	25.9	23.8	7.2	21.4	10.7	27.5	24.7	26.0
	VSM-SingOnly (This work)	30.8	36.9	33.6	9.8	34.4	15.2	39.5	35.7	37.5
	VSM-SingPairMix (This work)	27.0	46.5	34.2	9.0	42.1	14.9	34.0	45.4	38.9
	BERT-SingOnly (This work)	**35.3**	41.9	38.3	9.8	32.5	15.1	44.0	38.6	41.1
	BERT-SingPairMix (This work)	33.6	**67.1**	**44.8**	**13.6**	**70.2**	**22.8**	**44.7**	**68.0**	**53.9**
XSum	LEAD-Baseline	8.5	9.4	8.9	5.3	9.5	6.8	13.8	9.4	11.2
	SumBasic [138]	8.7	9.7	9.2	5.0	8.9	6.4	13.7	9.4	11.1
	KL-Summ (Haghighi et al., 2009)	9.2	10.2	9.7	5.0	8.9	6.4	14.2	9.7	11.5
	LexRank [139]	9.7	10.8	10.2	5.5	9.8	7.0	15.2	10.4	12.4
	VSM-SingOnly (This work)	12.3	14.1	13.1	3.8	11.0	5.6	17.9	12.0	14.4
	VSM-SingPairMix (This work)	10.1	22.6	13.9	4.2	17.4	6.8	14.3	20.8	17.0
	BERT-SingOnly (This work)	24.2	26.1	25.1	6.6	16.7	9.5	35.3	20.8	26.2
	BERT-SingPairMix (This work)	**33.2**	**56.0**	**41.7**	**24.1**	**65.5**	**35.2**	**57.3**	**59.6**	**58.5**
DUC-04	LEAD-Baseline	6.0	4.8	5.3	2.8	3.8	3.2	8.8	4.4	5.9
	SumBasic [138]	4.2	3.2	3.6	3.0	3.8	3.3	7.2	3.4	4.6
	KL-Summ (Haghighi et al., 2009)	5.6	4.5	5.0	2.8	3.8	3.2	8.0	4.2	5.5
	LexRank [139]	8.5	6.7	7.5	**4.8**	6.5	5.5	12.1	6.6	8.6
	VSM-SingOnly (This work)	**18.0**	**14.7**	**16.2**	3.6	8.4	5.0	**23.6**	**11.8**	**15.7**
	VSM-SingPairMix (This work)	3.8	6.2	4.7	3.6	11.4	5.5	7.4	8.0	7.7
	BERT-SingOnly (This work)	8.4	6.5	7.4	2.8	5.3	3.7	15.6	6.6	9.2
	BERT-SingPairMix (This work)	4.8	9.1	6.3	4.2	**14.2**	**6.5**	9.0	10.9	9.9

Instance selection results; evaluated for primary, secondary, and all ground-truth sentences. Our BERT-SingPairMix method achieves strong performance owing to its capability of building effective representations for both singletons and pairs.

Further, we include the LEAD method that selects the first N sentences from each document. We then require all systems to extract N instances, i.e., either singletons or pairs, from the input document(s).[6]

We compare system-identified instances with ground-truth instances, and in particular, we compare against the primary, secondary, and full set of ground-truth sentences. A *primary* sentence is defined as a ground-truth singleton or a sentence in a ground-truth pair that has the highest similarity to the reference summary sentence; the other sentence in the pair is considered *secondary*, which provides complementary information to the primary sentence. E.g., let S^*={(1, 2), 5, (8, 4), 10} be a ground-truth set of instances, where numbers are sentence indices and the first sentence of each pair is primary. Our ground-truth primary set thus contains {1, 5, 8, 10}; secondary set contains {2, 4}; and the full set of ground-truth sentences contains {1, 2, 5, 8, 4, 10}. Assume S={(1, 2), 3, (4, 10), 15} are system-selected instances. We uncollapse all pairs to obtain a set of single sentences S={1, 2, 3, 4, 10, 15}, then compare them against the primary, secondary, and full set of ground-truth sentences to calculate precision, recall, and F1-measure scores. This evaluation scheme allows a fair comparison of a variety of systems for instance selection, and assess their performance on identifying primary and secondary sentences respectively for summary generation.

Extraction Results In Table 3.2 we present instance selection results for the CNN/DM, XSum, and DUC-04 datasets. Our method builds representations for instances using either BERT or VSM (§3.1.2.1). To ensure a thorough comparison, we experiment with selecting a mixed set of singletons and pairs ("SingPairMix") as well as selecting singletons only ("SingOnly"). On the CNN/DM and XSum datasets, we observe that selecting a mixed set of singletons and pairs based on BERT representations (BERT+SingPairMix) demonstrates the most competitive results. It outperforms a number of strong baselines when evaluated on a full set of ground-truth sentences. The method also performs superiorly on identifying secondary sentences. For example, it increases recall scores for identifying secondary sentences from 33.8% to 69.8% (CNN/DM) and from 16.7% to 65.3% (XSum). Our method is able to achieve strong performance on instance selection owing

[6]We use N=4/1/5 respectively for the CNN/DM, XSum, and DUC-04 datasets. N is selected as the average number of sentences in reference summaries.

to BERT's capability of building effective representations for both singletons and pairs. It learns to identify salient source content based on token and position embeddings and it encodes sentential semantic compatibility using the pretraining task of predicting the next sentence; both are valuable additions to summary instance selection.

Further, we observe that identifying summary-worthy singletons and pairs from multi-document inputs (DUC-04) appears to be more challenging than that of single-document inputs (XSum and CNN/DM). This distinction is not surprising given that for multi-document inputs, the system has a large and diverse search space where candidate singletons and pairs are gathered from a set of documents written by different authors.[7] We find that the BERT model performs consistently on identifying secondary sentences, and VSM yields considerable performance gain on selecting primary sentences. Both BERT and VSM models are trained on the CNN/DM dataset and applied to DUC-04 as the latter data are only used for testing. Our findings suggest that the TF-IDF features of the VSM model are effective for multi-document inputs, as important topic words are usually repeated across documents and TF-IDF scores can reflect topical importance of words. This analysis further reveals that extending BERT to incorporate topical salience of words can be a valuable line of research for future work.

Summarization Results We present summarization results in Table 3.3, where we assess both extractive and abstractive summaries generated by BERT-SingPairMix. We omit VSM results as they are not as competitive as BERT on instance selection for the mixed set of singletons and pairs. The extractive summaries "BERT-Extr" are formed by concatenating selected singletons and pairs for each document, whereas "GT-SingPairMix" concatenates *ground-truth* singletons and pairs; it provides an upper bound for any system generating a set of singletons and pairs as the summary. To assure fair comparison, we limit all extractive summaries to contain up to 100 words (40 words for XSum) for ROUGE evaluation[8], where R-1, R-2, R-L, and R-SU4 are variants used to measure the overlap of unigrams, bigrams, longest common subsequences, and skip bigrams

[7]For the DUC-04 dataset, we select top K sentences from each document (K=5) and pool them as candidate singletons. Candidate pairs consist of arbitrary combinations of singletons. For all datasets we perform downsampling to balance the number of positive and negative singletons (or pairs).

[8]w/ ROUGE options: -n 2 -m -2 4 -w 1.2 -c 95 -r 1000 -l 100

Table 3.3: Summarization results on various datasets.

	CNN/Daily Mail		
System	**R-1**	**R-2**	**R-L**
SumBasic [138]	34.11	11.13	31.14
KLSumm (Haghighi et al., 2009)	29.92	10.50	27.37
LexRank [139]	35.34	13.31	31.93
PointerGen+Cov [11]	39.53	17.28	36.38
BERT-Abs w/ SS (This Work)	35.49	15.12	33.03
BERT-Abs w/ PG (This Work)	37.15	15.22	34.60
BERT-Extr (This Work)	**41.13**	**18.68**	**37.75**
GT-SingPairMix (This Work)	48.73	26.59	45.29
	XSum		
System	**R-1**	**R-2**	**R-L**
SumBasic [138]	18.56	2.91	14.88
KLSumm (Haghighi et al., 2009)	16.73	2.83	13.53
LexRank [139]	17.95	3.00	14.30
BERT-Abs w/ PG (This Work)	**25.08**	**6.48**	**19.75**
BERT-Extr (This Work)	23.53	4.54	17.23
GT-SingPairMix (This Work)	27.90	7.31	21.04
	DUC-04		
System	**R-1**	**R-2**	**R-SU4**
SumBasic [138]	29.48	4.25	8.64
KLSumm (Haghighi et al., 2009)	31.04	6.03	10.23
LexRank [139]	34.44	7.11	11.19
Extract+Rewrite [55]	28.90	5.33	8.76
Opinosis [69]	27.07	5.03	8.63
BERT-Abs w/ PG (This Work)	27.95	4.13	7.75
BERT-Extr (This Work)	30.49	5.12	9.05
GT-SingPairMix (This Work)	41.42	13.67	16.38

Whether abstractive summaries (BERT-Abs) outperform its extractive variant (BERT-Extr) appears to be related to the amount of sentence pairs selected by BERT-SingPairMix. Selecting more pairs than singletons seems to hurt the abstractor.

(with a maximum distance of 4) between system and reference summaries [97]. The abstractive summaries are generated from the same singletons and pairs used to form system extracts. "BERT-Abs-PG" generates an abstract by iteratively encoding singletons or pairs and decoding summary sentences using pointer-generator networks (§3.1.2.2).[9]

Our BERT summarization systems achieve results largely on par with those of prior work. It is interesting to observe that the extractive variant (BERT-Extr) can outperform its abstractive coun-

[9]We include an additional in-house system "BERT-Abs-SS" for CNN/DM that takes the same input but generates summary sentences using a tree-based decoder.

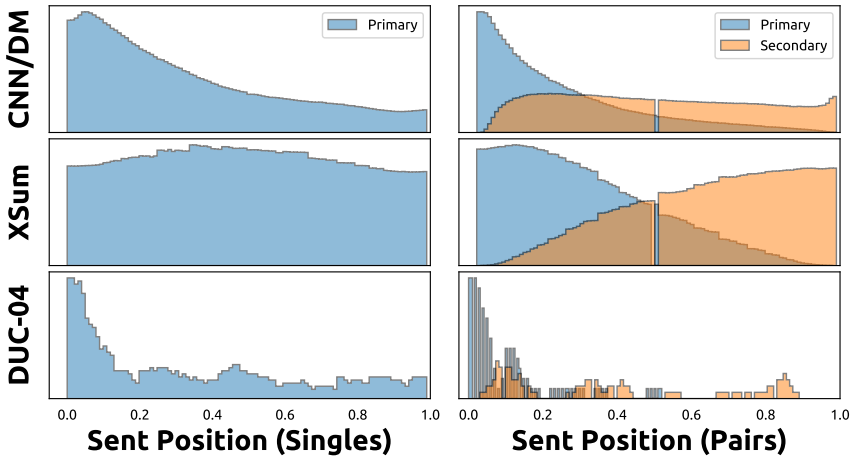

Figure 3.3: Position of ground-truth singletons and pairs in a document. The singletons of XSum can occur anywhere; the first and second sentence of a pair also appear far apart.

terparts on DUC-04 and CNN/DM datasets, and vice versa on XSum. A close examination of the results reveals that whether abstractive summaries outperform appears to be related to the amount of sentence pairs selected by "BERT-SingPairMix." Selecting more pairs than singletons seems to hurt the abstractor. For example, BERT selects 100% and 76.90% sentence pairs for DUC-04 and CNN/DM respectively, and 28.02% for XSum. These results suggest that existing abstractors using encoder-decoder models may need to improve on sentence fusion. These models are trained to generate fluent sentences more than preserving salient source content, leading to important content words being skipped in generating summary sentences. Our work intends to separate the tasks of sentence selection and summary generation, thus holding promise for improving compression and merging in the future. We present example system summaries in the supplementary.

Further analysis In this section we perform a series of analyses to understand where summary-worthy content is located in a document and how humans order them into a summary. Figure 3.3 shows the position of ground-truth singletons and pairs in a document. We observe that singletons of CNN/DM and DUC-04 tend to occur at the beginning of a document, whereas singletons of XSum can occur anywhere. We also find that the first and second sentence of a pair can appear far apart for XSum, but are closer for CNN/DM. These findings suggest that select-

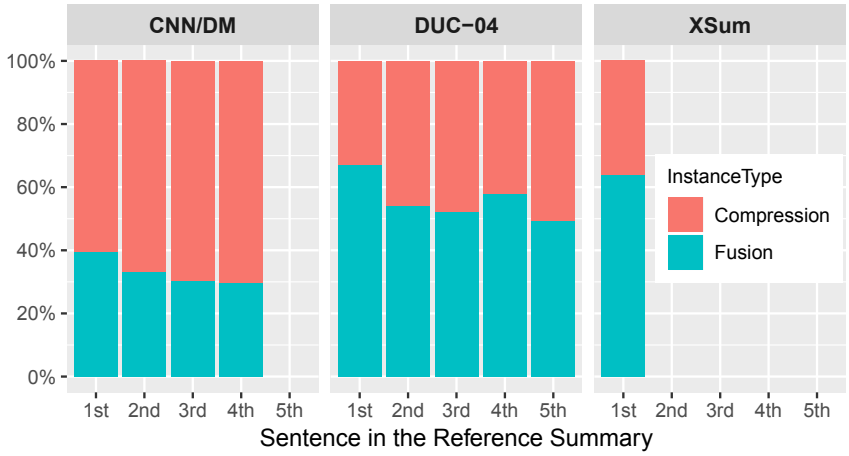

Figure 3.4: A sentence's *position* in a human summary can affect whether or not it is created by compression or fusion.

ing singletons and pairs for XSum can be more challenging than others, as indicated by the name "extreme" summarization.

Figure 3.4 illustrates how humans choose to organize content into a summary. Interestingly, we observe that a sentence's *position* in a human summary affects whether or not it is created by compression or fusion. The first sentence of a human-written summary is more likely than the following sentences to be a fusion of multiple source sentences. This is the case across all three datasets. We conjecture that the first sentence of a summary is expected to give an overview of the document and needs to consolidate information from different parts. Other sentences of a human summary can be generated by simply shortening singletons. Our statistics reveal that DUC-04 and XSum summaries involve more fusion operations, exhibiting a higher level of abstraction than CNN/DM.

3.1.5 Ground-truth Sets of Instances

We performed a manual inspection over a subset of our ground-truth sets of singletons and pairs. Each sentence from a human-written summary is matched with one or two source sentences based

Table 3.4: Sample ground-truth labels (CNN/DM)

Selected Source Sentence(s)	Human Summary Sentence
an inmate housed on the " forgotten floor , " where many mentally ill inmates are housed in miami before trial .	mentally ill inmates in miami are housed on the " forgotten floor "
most often , they face drug charges or charges of assaulting an officer – charges that judge steven leifman says are usually " avoidable felonies . "	judge steven leifman says most are there as a result of " avoidable felonies "
" i am the son of the president . miami , florida -lrb- cnn -rrb- – the ninth floor of the miami-dade pretrial detention facility is dubbed the " forgotten floor . "	while cnn tours facility , patient shouts : " i am the son of the president "
it 's brutally unjust , in his mind , and he has become a strong advocate for changing things in miami . so , he says , the sheer volume is overwhelming the system , and the result is what we see on the ninth floor .	leifman says the system is unjust and he 's fighting for change .

Selected Source Sentence(s)	Human Summary Sentence
the average surface temperature has warmed one degree fahrenheit -lrb- 0.6 degrees celsius -rrb- during the last century , according to the national research council .	earth has warmed one degree in past 100 years .
the reason most cited – by scientists and scientific organizations – for the current warming trend is an increase in the concentrations of greenhouse gases , which are in the atmosphere naturally and help keep the planet 's temperature at a comfortable level . in the worst-case scenario , experts say oceans could rise to overwhelming and catastrophic levels , flooding cities and altering seashores .	majority of scientists say greenhouse gases are causing temperatures to rise .
a change in the earth 's orbit or the intensity of the sun 's radiation could change , triggering warming or cooling . other scientists and observers , a minority compared to those who believe the warming trend is something ominous , say it is simply the latest shift in the cyclical patterns of a planet 's life .	some critics say planets often in periods of warming or cooling .

Sample of our ground-truth labels for singleton/pair instances from CNN/Daily Mail. Large chunks of text are copied straight out of the source sentences.

on average ROUGE similarity (details in Section 3.1.3). Tables 3.4, 3.5, and 3.6 present randomly selected examples from CNN/Daily Mail, XSum, and DUC-04, respectively. Colored text represents overlapping tokens between sentences. Darker colors represent content from primary sentences, while lighter colors represent content from secondary sentences. Best viewed in color.

3.1.6 Example Summaries

Table 3.7 presents example system summaries and human-written abstracts from CNN/Daily Mail. Each Human Abstract sentence is matched with a sentence singleton or pair from the source document; these singletons/pairs make up the GT-SingPairMix summary. Similarly, each sentence from BERT-Abs is created by compressing a singleton or merging a pair selected by BERT-Extr.

Table 3.5: Sample of our ground-truth labels (XSum)

Selected Source Sentence(s)	Human Summary Sentence
the premises , used by east belfast mp naomi long , have been targeted a number of times . army explosives experts were called out to deal with a suspect package at the offices on the newtownards road on friday night .	a suspicious package left outside an alliance party office in east belfast has been declared a hoax .

Selected Source Sentence(s)	Human Summary Sentence
nev edwards scored an early try for sale , before castres ' florian vialelle went over , but julien dumora 's penalty put the hosts 10-7 ahead at the break .	a late penalty try gave sale victory over castres at stade pierre-antoine in their european challenge cup clash .

Selected Source Sentence(s)	Human Summary Sentence
speaking in the dáil , sinn féin leader gerry adams also called for a commission of investigation and said his party had " little confidence the government is protecting the public interest " . last year , nama sold its entire 850-property loan portfolio in northern ireland to the new york investment firm cerberus for more than # 1bn .	the irish government has rejected calls to set up a commission of investigation into the sale of nama 's portfolio of loans in northern ireland .

Sample of our ground-truth labels for singleton/pair instances from XSum. Each article has only one summary sentences, and thus only one singleton or pair matched with it.

3.1.7 Conclusion

We present an investigation into the feasibility of scoring singletons and pairs according to their likelihoods of producing summary sentences. Our framework is founded on the human process of selecting one or two sentences to merge together and it has the potential to bridge the gap between compression and fusion studies. Our method provides a promising avenue for domain-specific summarization where content selection and summary generation are only loosely connected to reduce the costs of obtaining massive annotated data.

3.2 A Cascade Approach to Content Selection for Neural Abstractive Summarization

We present an empirical study in favor of a cascade architecture to neural text summarization. Summarization practices vary widely but few other than news summarization can provide a sufficient amount of training data enough to meet the requirement of end-to-end neural abstractive systems which perform content selection and surface realization jointly to generate abstracts. Such

Table 3.6: Sample of our ground-truth labels (DUC-04)

Selected Source Sentence(s)	Human Summary Sentence
hun sen 's cambodian people 's party won 64 of the 122 parliamentary seats in july 's elections , short of the two-thirds majority needed to form a government on its own .	cambodian elections , fraudulent according to opposition parties , gave the cpp of hun sen a scant majority but not enough to form its own government .
opposition leaders prince norodom ranariddh and sam rainsy , citing hun sen 's threats to arrest opposition figures after two alleged attempts on his life , said they could not negotiate freely in cambodia and called for talks at sihanouk 's residence in beijing . cambodian leader hun sen has guaranteed the safety and political freedom of all politicians , trying to ease the fears of his rivals that they will be arrested or killed if they return to the country .	opposition leaders fearing arrest , or worse , fled and asked for talks outside the country .
the cambodian people 's party criticized a non-binding resolution passed earlier this month by the u.s. house of representatives calling for an investigation into violations of international humanitarian law allegedly committed by hun sen .	the un found evidence of rights violations by hun sen prompting the us house to call for an investigation .
cambodian politicians expressed hope monday that a new partnership between the parties of strongman hun sen and his rival , prince norodom ranariddh , in a coalition government would not end in more violence .	the three-month governmental deadlock ended with han sen and his chief rival , prince norodom ranariddh sharing power .
citing hun sen 's threats to arrest opposition politicians following two alleged attempts on his life , ranariddh and sam rainsy have said they do not feel safe negotiating inside the country and asked the king to chair the summit at gis residence in beijing . after a meeting between hun sen and the new french ambassador to cambodia , hun sen aide prak sokhonn said the cambodian leader had repeated calls for the opposition to return , but expressed concern that the international community may be asked for security guarantees .	han sen guaranteed safe return to cambodia for all opponents but his strongest critic , sam rainsy , remained wary .
diplomatic efforts to revive the stalled talks appeared to bear fruit monday as japanese foreign affairs secretary of state nobutaka machimura said king norodom sihanouk has called on ranariddh and sam rainsy to return to cambodia . king norodom sihanouk on tuesday praised agreements by cambodia 's top two political parties – previously bitter rivals – to form a coalition government led by strongman hun sen .	chief of state king norodom sihanouk praised the agreement .

Sample of our ground-truth labels for singleton/pair instances from DUC-04, a multi-document dataset. Ground-truth sentences are widely dispersed among all ten documents.

Table 3.7: Example system summaries and human-written abstracts.

Extractive Upper Bound	**Human Abstract**
• She's a high school freshman with Down syndrome. • Trey – a star on Eastern High School's basketball team in Louisville, Kentucky, who's headed to play college ball next year at Ball State – was originally going to take his girlfriend to Eastern's prom. • Trina Helson, a teacher at Eastern, alerted the school's newspaper staff to the prom-posal and posted photos of Trey and Ellie on Twitter that have gone viral.	• College-bound basketball star asks girl with Down syndrome to high school prom. • Pictures of the two during the "prom-posal" have gone viral.
BERT-Extractive	**BERT-Abstractive**
• But all that changed Thursday when Trey asked Ellie to be his prom date. • Trey – a star on Eastern High School's basketball team in Louisville, Kentucky, who's headed to play college ball next year at Ball State – was originally going to take his girlfriend to Eastern's prom. • Trina Helson, a teacher at Eastern, alerted the school's newspaper staff to the prom-posal and posted photos of Trey and Ellie on Twitter that have gone viral. • (CNN) He's a blue chip college basketball recruit. • She's a high school freshman with Down syndrome.	• Trey asked Ellie to be his prom date. • Trina Helson, a teacher at Eastern, alerted the school's newspaper staff. • He's a high school student with Down syndrome.

Extractive Upper Bound	**Human Abstract**
• Marseille prosecutor Brice Robin told CNN that "so far no videos were used in the crash investigation." • Reichelt told "Erin Burnett: outfront" that he had watched the video and stood by the report, saying Bild and Paris Match are "very confident" that the clip is real. • Lubitz told his Lufthansa flight training school in 2009 that he had a "previous episode of severe depression," the airline said Tuesday.	• Marseille prosecutor says "so far no videos were used in the crash investigation" despite media reports. • Journalists at Bild and Paris Match are "very confident" the video clip is real, an editor says. • Andreas Lubitz had informed his Lufthansa training school of an episode of severe depression, airline says.
BERT-Extractive	**BERT-Abstractive**
• Marseille, France (CNN) - the French prosecutor leading an investigation into the crash of Germanwings flight 9525 insisted Wednesday that he was not aware of any video footage from on board the plane. • Marseille prosecutor Brice Robin told CNN that "so far no videos were used in the crash investigation." • Robin's comments follow claims by two magazines, German Daily Bild and French Paris Match, of a cell phone video showing the harrowing final seconds from on board Germanwings flight 9525 as it crashed into the French Alps. • The two publications described the supposed video, but did not post it on their websites.	• New : French prosecutor says he was not aware of video footage from on board the plane. • Two magazines, including German Daily Bild, have been described as the video.

Each Human Abstract sentence is lined up horizontally with its corresponding ground-truth instance, which is found in Extractive Upper Bound summary. Similarly, each sentence from BERT-Abstractive is lined up horizontally with its corresponding instance selected by BERT-Extractive. The sentences are manually de-tokenized for readability.

systems also pose a challenge to summarization evaluation, as they force content selection to be evaluated along with text generation, yet evaluation of the latter remains an unsolved problem. In this section, we present empirical results showing that the performance of a cascaded pipeline that separately identifies important content pieces and stitches them together into a coherent text is comparable to or outranks that of end-to-end systems, whereas a pipeline architecture allows for flexible content selection. We finally discuss how we can take advantage of a cascaded pipeline in neural text summarization and shed light on important directions for future research.[10]

3.2.1 Introduction

There is a variety of successful summarization applications but few can afford to have a large number of annotated examples that are sufficient to meet the requirement of end-to-end neural abstractive summarization. Examples range from summarizing radiology reports [59, 60] to congressional bills [61] and meeting conversations [62, 63, 64]. The lack of annotated resources suggests that end-to-end systems may not be a "one-size-fits-all" solution to neural text summarization. There is an increasing need to develop cascaded architectures to allow for customized content selectors to be combined with general-purpose neural text generators to realize the full potential of neural abstractive summarization.

We advocate for explicit content selection as it allows for a rigorous evaluation and visualization of intermediate results of such a module, rather than associating it with text generation. Existing neural abstractive systems can perform content selection implicitly using end-to-end models [11, 12, 13, 14], or more explicitly, with an external module to select important sentences or words to aid generation [9, 18, 46, 22, 49, 133, 140, 50]. However, content selection concerns not only the selection of important segments from a document, but also the cohesiveness of selected

[10]This section is adapted from: L. Lebanoff, F. Dernoncourt, D. S. Kim, W. Chang, and F. Liu, A Cascade Approach to Content Selection for Neural Abstractive Summarization, in Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing (AACL-IJCNLP), 2020.

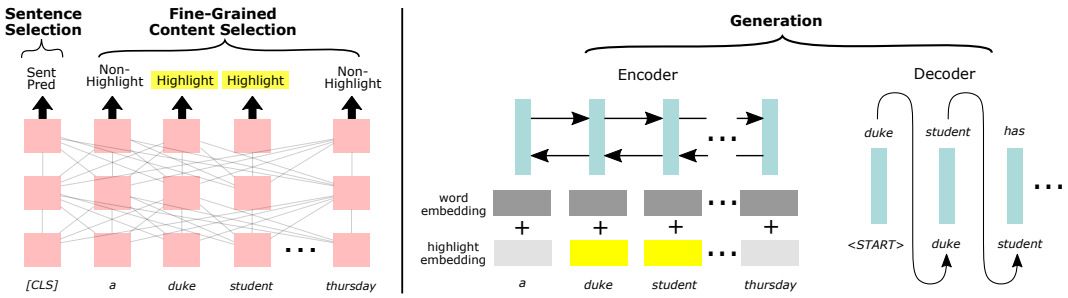

Figure 3.5: Model architecture. We divide the task between two main components: the first component performs sentence selection and fine-grained content selection, which are posed as a classification problem and a sequence-tagging problem, respectively. The second component receives the first component's outputs as supplementary information to generate the summary. A cascade architecture provides the necessary flexibility to separate content selection from surface realization in abstractive summarization.

segments and the amount of text to be selected in order for a neural text generator to produce a summary.

In this section, we aim to investigate the feasibility of a cascade approach to neural text summarization. We explore a constrained summarization task, where an abstract is created one sentence at a time through a cascaded pipeline. Our pipeline architecture chooses one or two sentences from the source document, then highlights their summary-worthy segments and uses those as a basis for composing a summary sentence. When a pair of sentences are selected, it is important to ensure that they are *fusible*—there exists cohesive devices that tie the two sentences together into a coherent text—to avoid generating nonsensical outputs [65, 66]. Highlighting sentence segments allows us to perform fine-grained content selection that guides the neural text generator to stitch selected segments into a coherent sentence. The contributions of this work are summarized as follows.

- We present an empirical study in favor of a cascade architecture for neural text summarization. Our cascaded pipeline chooses one or two sentences from the document and highlights their important segments; these segments are passed to a neural generator to produce a summary sentence.
- Our quantitative results show that the performance of a cascaded pipeline is comparable to or outranks that of end-to-end systems, with added benefit of flexible content selection. We

discuss how we can take advantage of a cascade architecture and shed light on important directions for future research.[11]

3.2.2 A Cascade Approach

Our cascaded summarization approach focuses on shallow abstraction. It makes use of text transformations such as sentence shortening, paraphrasing and fusion [141] and is in contrast to deep abstraction, where a full semantic analysis of the document is often required. A shallow approach helps produce abstracts that convey important information while, crucially, remaining faithful to the original. In what follows, we describe our approach to select single sentences and sentence pairs from the document, highlight summary-worthy segments and perform summary generation conditioned on highlights.

Selection of Singletons and Pairs Our approach iteratively selects one or two sentences from the input document; they serve as the basis for composing a single summary sentence. Previous research suggests that 60-85% of human-written summary sentences are created by shortening a single sentence or merging a pair of sentences [140]. We adopt this setting and present a coarse-to-fine strategy for content selection. Our strategy begins with selecting sentence singletons and pairs, followed by highlighting important segments of the sentences. Importantly, the strategy allows us to control which segments will be combined into a summary sentence—"compatible" segments come from either a single document sentence or a pair of *fusible* sentences. In contrast, when all important segments of the document are provided to a neural generator all at once [18], it can happen that the generator arbitrarily stitches together text segments from unrelated sentences, yielding a summary that contains hallucinated content and fails to retain the meaning of the original document [21, 26, 25].

We expect a sentence singleton or pair to be selected from the document if it contains salient content. Moreover, a pair of sentences should contain content that is compatible with each other.

[11] Our code is publicly available at `https://github.com/ucfnlp/cascaded-summ`

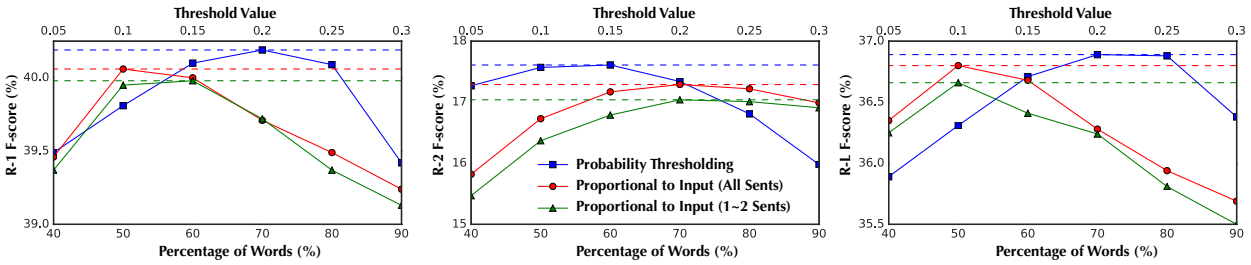

Figure 3.6: Comparison of various highlighting strategies. Thresholding obtains the best performance.

Given a sentence or pair of sentences from the document, our model predicts whether it is a valid instance to be compressed or merged to form a summary sentence. We follow [140] to use BERT [47] to perform the classification. BERT is a natural choice since it takes one or two sentences and generates a classification prediction. It treats an input singleton or pair of sentences as a sequence of tokens. The tokens are fed to a series of Transformer block layers, consisting of multi-head self-attention modules. The first Transformer layer creates a contextual representation for each token, and each successive layer further refines those representations. An additional [CLS] token is added to contain the sentence representation. BERT is fine-tuned for our task by adding an output layer on top of the final layer representation $\mathbf{h}^L_{[\text{CLS}]}$ for sequence s, as seen in Eq. (3.6).

$$p_{\text{sent}}(s) = \sigma(\mathbf{u}^\top \mathbf{h}^L_{[\text{CLS}]}) \tag{3.6}$$

where $\mathbf{u}$ is a vector of weights and σ is the sigmoid function. The model predicts p_{sent} – whether the sentence singleton or pair is an appropriate one based on the [CLS] token representation. We describe the training data for this task in §5.2.6.

Fine-Grained Content Selection It is interesting to note that the previous architecture can be naturally extended to perform fine-grained content selection by highlighting important words of sentences. When two sentences are selected to generate a fusion sentence, it is desirable to identify segments of text from these sentences that are potentially compatible with each other. The coarse-to-fine method allows us to examine the intermediate results and compare them with ground-truth. Concretely, we add a classification layer to the final layer representation $\mathbf{h}^L_i$ for each token w_i (Eq. (3.7)). The per-target-word loss is then interpolated with instance prediction (one or two

sentences) loss using a coefficient λ. Such a multi-task learning objective has been shown to improve performance on a number of tasks [56].

$$p_{\text{highlight}}(w_i) = \sigma(\mathbf{v}^\top \mathbf{h}_i^L) \tag{3.7}$$

where $\mathbf{v}$ is a vector of weights and σ is the sigmoid function. The model predicts $p_{\text{highlight}}$ for each token – whether the token should be included in the output fusion, calculated based on the given token's representation.

Information Fusion Given one or two sentences taken from a document and their fine-grained highlights, we proceed by describing a fusion process that generates a summary sentence from the selected content. Our model employs an encoder-decoder architecture based on pointer-generator networks that has shown strong performance on its own and with adaptations [11, 18]. We feed the sentence singleton or pair to the encoder along with highlights derived by the fine-grained content selector, the latter come in the form of binary tags. The tags are transformed to a "*highlight-on*" embedding for each token if it is chosen by the content selector, and a "*highlight-off*" embedding for each token not chosen. The highlight-on/off embeddings are added to token embeddings in an element-wise manner; both highlight and token embeddings are learned. An illustration is shown in Figure 3.5.

Highlights provide a valuable intermediate representation suitable for shallow abstraction. Our approach thus provides an alternative to methods that use more sophisticated representations such as syntactic/semantic graphs [78, 142, 74]. It is more straightforward to incorporate highlights into an encoder-decoder fusion model, and obtaining highlights through sequence tagging can be potentially adapted to new domains.

Table 3.8: Summarization results.

System	R-1	R-2	R-L
SumBasic [138]	34.11	11.13	31.14
LexRank [139]	35.34	13.31	31.93
Pointer-Generator [11]	39.53	17.28	36.38
FastAbsSum [46]	40.88	17.80	**38.54**
BERT-Extr [140]	41.13	**18.68**	37.75
BottomUp [18]	**41.22**	**18.68**	38.34
BERT-Abs [140]	37.15	15.22	34.60
Cascade-Fusion (Ours)	40.10	17.61	**36.71**
Cascade-Tag (Ours)	**40.24**	**18.33**	36.14
GT-Sent + Sys-Tag	50.40	27.74	46.25
GT-Sent + Sys-Tag + Fusion	51.33	28.08	47.50
GT-Sent + GT-Tag	**74.80**	48.21	**67.40**
GT-Sent + GT-Tag + Fusion	72.70	**48.33**	67.06

(SYSTEM SENTS) A Duke student has admitted to hanging a noose made of rope from a tree near a student union, university officials said Thursday. The student was identified during an investigation by campus police and the office of student affairs and admitted to placing the noose on the tree early Wednesday, the university said.
(CASCADE-FUSION) A Duke student was identified during an investigation by campus police and the office of student affairs and admitted to placing the noose on the tree early Wednesday.

(GT SENTS) In a news release, it said the student was no longer on campus and will face student conduct review. Duke University is a private college with about 15,000 students in Durham, North Carolina.
(GT SENTS + FUSION) Duke University student was no longer on campus and will face student conduct review.
(REFERENCE) Student is no longer on Duke University campus and will face disciplinary review.

(LEFT) Summarization results on CNN/DM test set. Our cascade approach performs comparable to strong extractive and abstractive baselines; oracle models using ground-truth sentences and segment highlights perform the best. (RIGHT) Example source sentences and their fusions. Dark highlighting is content taken from the first sentence, and light highlighting comes from the second. Our *Cascade-Fusion* approach effectively performs entity replacement by replacing "student" in the second sentence with "a Duke student" from the first sentence.

3.2.3 Experimental Results

Data and Annotation To enable direct comparison with end-to-end systems, we conduct experiments on the widely used CNN/DM dataset [11] to report results of our cascade approach. We use the procedure described in Lebanoff et al. Lebanoff:2019 to create training instances for the sentence selector and fine-grained content selector. Our training data contains 1,053,993 instances; every instance contains one or two candidate sentences. It is a positive instance if a ground-truth summary sentence can be formed by compressing or merging sentences of the instance, negative otherwise. For positive instances, we highlight all lemmatized unigrams appearing in the summary, excluding punctuation. We further add smoothing to the labels by highlighting single words that connect two highlighted phrases and by dehighlighting isolated stopwords. At test time, four highest-scored instances are selected per document; their important segments are highlighted by content selector then passed to the fusion step to produce a summary sentence each. The hyperparameter λ for weighing the per-target-word loss is set to 0.2 and highlighting threshold value is 0.15. The model hyperparameters are tuned on the validation split.

Summarization Results We show experimental results on the standard test set and evaluated by ROUGE metrics [97] in Table 3.8. The performance of our cascade approaches, *Cascade-Fusion* and *Cascade-Tag*, is comparable to or outranks a number of extractive and abstractive baselines. Particularly, *Cascade-Tag* does not use a fusion step (§3.2.2) and is the output of fine-grained content selection. *Cascade-Fusion* provides a direct comparison against BERT-Abs [140] that uses sentence selection and fusion but lacks a fine-grained content selector.

Our results suggest that a coarse-to-fine content selection strategy remains necessary to guide the fusion model to produce informative sentences. We observe that the addition of the fusion model has only a moderate impact on ROUGE scores, but the fusion process can reorder text segments to create true and grammatical sentences, as shown in Table 3.8. We analyze the performance of a number of oracle models that use ground-truth sentence selection (GT-Sent) and tagging (GT-Tag). When given ground-truth sentences as input, our cascade models achieve $\sim$10

points of improvement in all ROUGE metrics. When the models are also given ground-truth highlights, they achieve an additional 20 points of improvement. In a preliminary examination, we observe that not all highlights are included in the summary during fusion, indicating there is space for improvement. These results show that cascade architectures have great potential to generate shallow abstracts and future emphasis may be placed on accurate content selection.

How much should we highlight? It is important to quantify the amount of highlighting required for generating a summary sentence. Highlighting too much or too little can be unhelpful. We experiment with three methods to determine the appropriate amount of words to highlight. *Probability Thresholding* chooses a set threshold whereby all words that have a probability higher than the threshold are highlighted. When *Proportional to Input* is used, the highest probability words are iteratively highlighted until a target rate is reached. The amount of highlighting can be proportional to the total number of words per instance (one or two sentences) or per document, containing all sentences selected for the document.

We investigate the effect of varying the amount of highlighting in Figure 3.6. Among the three methods, probability thresholding performs the best, as it gives more freedom to content selection. If the model scores all of the words in sentences highly, then we should correspondingly highlight all of the words. If only very few words score highly, then we should only pick those few.

Highlighting a certain percentage of words tend to perform less well. On our dataset, a threshold value of 0.15–0.20 produces the best ROUGE scores. Interestingly, these thresholds end up highlighting 58–78% of the words of each sentence. Compared to what the generator was trained on, which had a median of 31% of each sentence highlighted, the system's rate of highlighting is higher. If the model's highlighting rate is set to be similar to that of the ground-truth, it yields much lower ROUGE scores (cf. threshold value of 0.3 in Figure 3.6). This observation suggests that the amount of highlighting can be related to the effectiveness of content selector and it may be better to highlight more than less.

3.2.4 Conclusion

We present a cascade approach to neural abstractive summarization that separates content selection from surface realization. Importantly, our approach makes use of text highlights as intermediate representation; they are derived from one or two sentences using a coarse-to-fine content selection strategy, then passed to a neural text generator to compose a summary sentence. A successful cascade approach is expected to accurately select sentences and highlight an appropriate amount of text, both can be customized for domain-specific tasks.

CHAPTER 4: SENTENCE FUSION

In the previous chapter, we covered the importance of content selection for summarization. However, content selection alone is not enough to form a high quality summary. Summaries must be coherent and grammatical, which is handled by a surface realization model. Recent methods resort to simply copying or compressing single sentences to form a summary sentences, but humans can effectively fuse multiple sentences together. In order for summarization models to approach the ability of humans to summarize, these models will need to be able to fuse sentences in a coherent, grammatical, and faithful manner.

In Section 4.1, we perform an analysis of state-of-the-art summarization systems on performing sentence fusion. We find that the systems frequently produce fusion sentences that are ungrammatical and unfaithful to the original document.

In Section 4.2, we introduce a new dataset of sentence fusion examples containing what we call *points of correspondence* between sentences. Points of correspondence are segments of text that represent what ties two sentences together. Our data can be useful to future research as a testbed for sentence fusion models, and the points of correspondence data can be analyzed to better understand how humans easily perform sentence fusion.

In Section 4.3, we present methods for fusing sentences together using points of correspondence. We show that two strategies can be incorporated into Transformer model architectures leading to improved summary quality.

4.1 Analyzing Sentence Fusion for Abstractive Summarization

While recent work in abstractive summarization has resulted in higher scores in automatic metrics, there is little understanding on how these systems combine information taken from multiple document sentences. In this section, we analyze the outputs of five state-of-the-art abstractive summarizers, focusing on summary sentences that are formed by sentence fusion. We ask assessors to judge the grammaticality, faithfulness, and method of fusion for summary sentences. Our analysis reveals that system sentences are mostly grammatical, but often fail to remain faithful to the original article. [1]

4.1.1 Introduction

Modern abstractive summarizers excel at finding and extracting salient content [11, 46, 12, 50]. However, one of the key tenets of summarization is consolidation of information, and these systems can struggle to combine content from multiple source texts, yielding output summaries that contain poor grammar and even incorrect facts. Truthfulness of summaries is a vitally important feature in order for summarization to be widely accepted in real-world applications [143, 20]. In this work, we perform an extensive analysis of summary outputs generated by state-of-the-art systems, examining features such as truthfulness to the original document, grammaticality, and method of how sentences are merged together. This work presents the first in-depth human evaluation of multiple diverse summarization models.

We differentiate between two methods of shortening text: sentence compression and sentence fusion. Sentence compression reduces the length of a *single* sentence by removing words or rephrasing parts of the sentence [116, 29, 144, 58, 31]. Sentence fusion reduces *two or more* sentences to one by taking content from each sentence and merging them together [76, 87, 28].

[1] This section is adapted from: L. Lebanoff, J. Muchovej, F. Dernoncourt, D. S. Kim, S. Kim, W. Chang, and F. Liu, "Analyzing Sentence Fusion in Abstractive Summarization," in Proceedings for the EMNLP 2019 Workshop on New Frontiers in Summarization, 2019.

Table 4.1: Comparison of state-of-the-art summarization systems.

System	ROUGE			Created By				Avg Summ
	R-1	R-2	R-L	Compress	Fuse	Copy	Fail	Sent Len
PG [11]	39.53	17.28	36.38	63.14	6.44	**30.24**	0.18	15.7
Novel [22]	40.19	17.38	37.52	71.25	19.77	5.39	3.59	11.8
Fast-Abs-RL [46]	40.88	17.80	**38.54**	**96.65**	0.83	2.21	0.31	15.6
Bottom-Up [18]	41.22	18.68	38.34	71.15	16.35	11.76	0.74	10.7
DCA [12]	**41.69**	**19.47**	37.92	64.11	23.96	7.07	4.86	14.5
Reference Summaries	-	-	-	60.65	**31.93**	1.36	**6.06**	**19.3**

Middle column describes how summary sentences are generated. *Compress*: single sentence is shortened. *Fuse*: multiple sentences are merged. *Copy*: sentence is copied word-for-word. *Fail*: did not find matching source sentences.

Compression is considered an easier task because unimportant clauses within the sentence can be removed while retaining the grammaticality and truth of the sentence [145]. In contrast, fusion requires selection of important content and stitching of that content in a grammatical and meaningful way. We focus on sentence fusion in this work.

We examine the outputs of five abstractive summarization systems on CNN/DailyMail [114] using human judgments. Particularly, we focus on summary sentences that involve sentence fusion, since fusion is the task that requires the most improvement. We analyze several dimensions of the outputs, including faithfulness to the original article, grammaticality, and method of fusion. We present three main findings:

- 38.3% of the system outputs introduce incorrect facts, while 21.6% are ungrammatical;
- systems often simply concatenate chunks of text when performing sentence fusion, while largely avoiding other methods of fusion like entity replacement;
- systems struggle to reliably perform complex fusion, as entity replacement and other methods result in incorrect facts 47–75% of the time.

4.1.2 Evaluation Setup

Evaluation of summarization systems relies heavily on automatic metrics. However, ROUGE [97] and other n-gram based metrics are limited in evaluation power and do not tell the whole story [146]. They often focus on informativeness, which misses out on important facets of summaries such as faithfulness and grammaticality. In this work we present a thorough investigation of several abstractive summarization systems using human evaluation on CNN/DailyMail. The task was accomplished via the crowdsourcing platform Amazon Mechanical Turk. We particularly focus on summary sentences formed by sentence fusion, as it is arguably a harder task and is a vital aspect of abstractive summarization.

4.1.2.1 Summarization Systems

We narrowed our evaluation to five state-of-the-art summarization models[2], as they represent some of the most competitive abstractive summarizers developed in recent years. The models show diversity across several dimensions, including ROUGE scores, abstractiveness, and training paradigm. We briefly describe each system, along with a comparison in Table 4.1.

- **PG** [11] The pointer-generator networks use an encoder-decoder architecture with attention and copy mechanisms that allow it to either generate a new word from the vocabulary or copy a word directly from the document. It tends strongly towards extraction and copies entire summary sentences about 30% of the time.
- **Novel** [22] This model uses an encoder-decoder architecture but adds a novelty metric which is optimized using reinforcement learning. It improves summary novelty by promoting the use of unseen words.

[2]The summary outputs from PG, Bottom-Up, and Fast-Abs-RL are obtained from their corresponding Github repos. Those from Novel and DCA are graciously provided to us by the authors. We thank the authors for sharing their work.

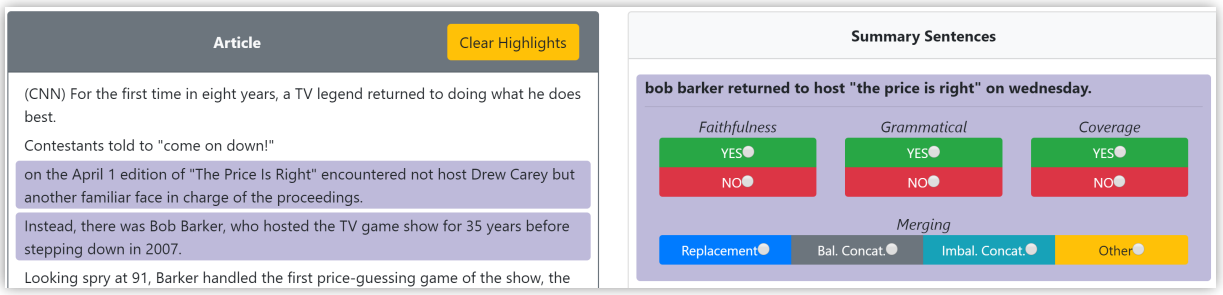

Figure 4.1: Annotation interface. A sentence from a random summarization system is shown along with four questions.

- **Fast-Abs-RL** [46] Document sentences are selected using reinforcement learning and then compressed/paraphrased using an encoder-decoder model to generate summary sentences.
- **Bottom-Up** [18] An external content selection model identifies which words from the document should be copied to the summary; such info is incorporated into the copy mechanism of an encoder-decoder model.
- **DCA** [12] The source text is divided among several encoders, which are all connected to a single decoder using hierarchical attention. It achieves one of the highest ROUGE scores among state-of-the-art.

4.1.2.2 Task Design

Our goal is to assess the quality of summary sentences according to their grammaticality, faithfulness and method of fusion. We design a crowd task consisting of a single article with six summary sentences: one sentence is guaranteed to be from the reference summary, the other five are taken from system summaries. An annotator is instructed to read the article, then rate the following characteristics for each summary sentence:

Faithfulness For a summary to be useful, it must remain true to the original text. This is particularly challenging for abstractive systems since they require a deep understanding of the document in order to rephrase sentences with the same meaning.

Grammaticality System summaries should follow grammatical rules in order to read well. Maintaining grammaticality can be relatively straightforward for sentence compression, as systems generally succeed at removing unnecessary clauses and interjections [11]. However, sentence fusion requires greater understanding in order to stitch together clauses in a grammatical way.

Method of Merging Each summary sentence in our experiments is created by fusing content from two document sentences. We would like to understand how this fusion is performed. The following possibilities are given:

- *Replacement:* a pronoun or description of an entity in one sentence is replaced by a different description of that entity in the other sentence.
- *Balanced concatenation:* a consecutive part of one sentence is concatenated with a consecutive part of the other sentence. The parts taken from each sentence are of similar length.
- *Imbalanced concatenation:* similar to the case of "balanced concatenation," but the part taken from one sentence is larger than the part taken from the other sentence.
- *Other:* all remaining cases.

Coverage An annotator is asked to rate how well highlighted article sentences "covered" the information contained in the summary sentence. Two article sentences that best match a summary sentence are selected according to a heuristic developed by [140]. The same heuristic is also used to determine whether a summary sentence is created by compression or fusion (more details later in this section). Given the importance of this heuristic for our task, we would like to measure its effectiveness on selecting article sentences that best match a given summary sentence.

We provide detailed instructions, including examples and explanations. We randomly select 100 articles from the CNN/DailyMail test set. This results in 100 tasks for annotators, where each task includes an article and six summary sentences to be evaluated—one of which originates from the reference summary and the other five are from any of the system summaries. Each task is completed by an average of 4 workers. All workers are required to have the "Master" qualification, a designation for high-quality annotations. Of the 600 summary sentences evaluated, each state-of-the-art system contributes as follows—*Bottom-Up*: 146, *DCA*: 130, *PG*: 37, *Novel*: 171, *Fast-Abs-RL*: 16, and *Reference*: 100. The number of sentences we evaluate for each system is proportional to the number of observed fusion cases.

In order to answer the *Method of Merging* and *Coverage* questions, the annotator must be provided with which two article sentences were fused together to create the summary sentence in question. We use the heuristic proposed by [140] to estimate which pair of sentences should be chosen. They use averaged ROUGE-1, -2, -L scores [97] to represent sentence similarity. The heuristic calculates the ROUGE similarity between the summary sentence and each article sentence. The article sentence with the highest similarity is chosen as the first sentence, then overlapping words are removed from the summary sentence. It continues to find the article sentence most similar to the remaining summary sentence, which is chosen as the second sentence. Our interface automatically highlights this pair of sentences (Figure 4.1).

The same heuristic is also employed in deciding whether a summary sentence was generated by sentence compression or fusion. The algorithm halts if no article sentence is found that shares two or more content words with the summary sentence. If it halts after only one sentence is found, then it is classified as *compression*. If it finds a second sentence, then it is classified as *fusion*.

4.1.3 Results

We present experimental results in Table 4.2. Our findings suggest that system summary sentences formed by fusion have low faithfulness (61.7% on average) as compared to the reference sum-

Table 4.2: Human evaluation results.

System	Faithful	Grammatical	Coverage
DCA	47.0	72.4	62.6
Bottom-Up	56.9	78.9	78.5
Novel	58.5	78.5	75.3
Fast-Abs-RL	69.0	77.6	82.8
PG	76.9	84.6	89.5
Reference	88.4	91.6	74.9

Percentage of summary sentences that are faithful, grammatical, etc. according to human evaluation of several state-of-the-art summarization systems (see §4.1.2 for details).

maries. This demonstrates the need for current summarization models to put more emphasis on improving the faithfulness of generated summaries. Surprisingly, the highest performing systems, DCA and Bottom-Up, according to ROUGE result in the lowest scores for being faithful to the article. While we cannot attribute the drop in faithfulness to an over-emphasis on optimizing automatic metrics, we can state that higher ROUGE scores does not necessarily lead to more faithful summaries, as other works have shown [21]. Bottom-Up, interestingly, is 20 points lower than PG, which it is closely based on. It uses an external content selector to choose what words to copy from the article. While identifying summary-worthy content improved ROUGE, we believe that Bottom-Up stitches together sections of content that do not necessarily belong together. Thus, it is important to identify not just summary-worthy content, but also *mergeable* content.

System summary sentences created by fusion are generally grammatical (78.4% on average), though it is still not up to par with reference summaries (91.6%). The chosen state-of-the-art systems use the encoder-decoder architecture, which employs a neural language model as the decoder, and language models generally succeed at encoding grammar rules and staying fluent [1]. The coverage for reference summaries is moderately high (74.9%), demonstrating the effectiveness of the heuristic of identifying where summary content is pulled from. Especially for most of the systems, the heuristic successfully finds the correct source sentences. As it is based mostly on word overlap,

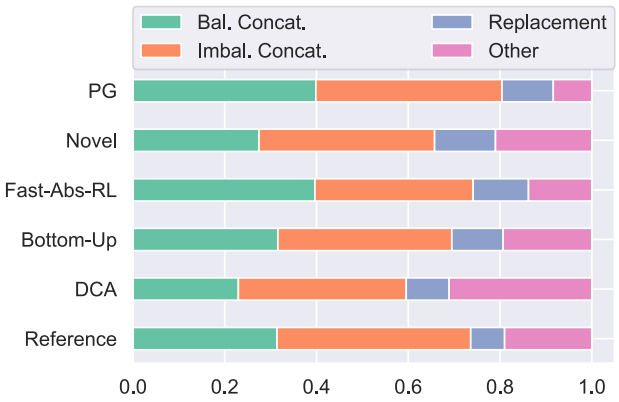

Figure 4.2: Frequency of each merging method. Concatenation is the most common method of merging.

the heuristic works better on summaries that are more extractive, hence the higher coverage scores among the systems compared to reference summaries, which are more abstractive.

Figure 4.2 illustrates the frequency of each merging method over the summarization systems. Most summary sentences are formed by concatenation. PG in particular most often fuses two sentences using concatenation. Surprisingly, very few reference summaries use entity replacement when performing fusion. We believe this is due to the extractiveness of the CNN/DailyMail dataset, and would likely have higher occurrences in more abstractive datasets.

Does the way sentences are fused affect their faithfulness and grammaticality? Table 4.3 provides insights regarding this question. Grammaticality is relatively high for all merging categories. Coverage is also high for balanced/imbalanced concatenation and replacement, meaning the heuristic works succesfully for these forms of sentence merging. It does not perform as well on the Other category. This is understandable, since sentences formed in a more complex manner will be harder to identify using simple word overlap. Faithfulness has a similar trend, with summaries generated using concatenation being more likely to be faithful to the original article. This may explain why PG is the most faithful of the systems, while being the simplest—it uses concatenation more than any of the other systems. We believe more effort can be directed towards improving the more complex merging paradigms, such as entity replacement.

Table 4.3: Results for each merging method.

System	Faithful	Grammatical	Coverage
Bal Concat	82.55	86.91	94.43
Imbal Concat	69.40	80.25	84.58
Replacement	53.06	82.04	77.55
Other	25.20	68.23	27.04

Concatenation has high faithfulness, grammaticality, and coverage, while Replacement and Other have much lower scores.

There are a few potential limitations associated with the experimental design. Judging whether a sentence is faithful to the original article can be a difficult task to perform reliably, even for humans. We observe that the reference summaries achieve lower than the expected faithfulness and grammaticality of 100%. This can have two reasons. First, the inter-annotator agreement for this task is relatively low and we counteract this by employing an average of four annotators to complete each task. Second, we make use of an automatic heuristic to highlight sentence pairs from the article. While it generally finds the correct sentences—average Coverage score of 77.3%—the incorrect pairs may have biased the annotators away from sentences that humans would have found more appropriate. This further exemplifies the difficulty of the task.

4.1.4 Conclusion

In this section we present an investigation into sentence fusion for abstractive summarization. Several state-of-the-art systems are evaluated, and we find that many of the summary outputs generate false information. Most of the false outputs were generated by entity replacement and other complex merging methods. These results demonstrate the need for more attention to be focused on improving sentence fusion and entity replacement.

4.2 Understanding Points of Correspondence between Sentences for Abstractive Summarization

Fusing sentences containing disparate content is a remarkable human ability that helps create informative and succinct summaries. Such a simple task for humans has remained challenging for modern abstractive summarizers, substantially restricting their applicability in real-world scenarios. In this section, we present an investigation into fusing sentences drawn from a document by introducing the notion of points of correspondence, which are cohesive devices that tie any two sentences together into a coherent text. The types of points of correspondence are delineated by text cohesion theory, covering pronominal and nominal referencing, repetition and beyond. We create a dataset containing the documents, source and fusion sentences, and human annotations of points of correspondence between sentences. Our dataset bridges the gap between coreference resolution and summarization. It is publicly shared to serve as a basis for future work to measure the success of sentence fusion systems.[3][4]

4.2.1 Introduction

Stitching portions of text together into a sentence is a crucial first step in abstractive summarization. It involves choosing which sentences to fuse, what content from each of them to retain and how best to present that information [88]. A major challenge in fusing sentences is to establish correspondence between sentences. If there exists no correspondence, it would be difficult, if not impossible, to fuse sentences. In Table 4.4, we present example source and fusion sentences, where the summarizer attempts to merge two sentences into a summary sentence with improper use

[3] `https://github.com/ucfnlp/points-of-correspondence`

[4] This section is adapted from: L. Lebanoff, J. Muchovej, F. Dernoncourt, D. S. Kim, L. Wang, W. Chang, and F. Liu, Understanding Points of Correspondence between Sentences for Abstractive Summarization, in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL): Student Research Workshop, 2020.

Table 4.4: Example unfaithful summary sentences.

[Source Sentences] Robert Downey Jr. is making headlines for walking out of an interview with a British journalist who dared to veer away from the superhero movie Downey was there to promote. The journalist instead started asking personal questions about the actor's political beliefs and "dark periods" of addiction and jail time.
[Summary] The journalist instead started asking personal questions about the actor's political beliefs
[Source Sentences] "Real Housewives of Beverly Hills" star and former child actress Kim Richards is accused of kicking a police officer after being arrested Thursday morning. A police representative said Richards was asked to leave but refused and then entered a restroom and wouldn't come out.
[Summary] Kim Richards is accused of kicking a police officer who refused to leave.
[Source Sentences] The kind of horror represented by the Blackwater case and others like it [...] may be largely absent from public memory in the West these days, but it is being used by the Islamic State in Iraq and Syria (ISIS) to support its sectarian narrative. In its propaganda, ISIS has been using Abu Ghraib and other cases of Western abuse to legitimize its current actions [...]
[Summary] In its propaganda, ISIS is being used by the Islamic State in Iraq and Syria.

Unfaithful summary sentences generated by neural abstractive summarizers, in-house and PG [11]. They attempt to merge two sentences into one sentence with improper use of *points of correspondence* between sentences, yielding nonsensical output. Summaries are manually re-cased for readability.

of *points of correspondence*. In this work, we seek to uncover hidden correspondences between sentences, which has a great potential for improving content selection and deep sentence fusion.

Sentence fusion (or multi-sentence compression) plays a prominent role in automated summarization and its importance has long been recognized [67]. Early attempts to fuse sentences build a dependency graph from sentences, then decode a tree from the graph using integer linear programming, finally linearize the tree to generate a summary sentence [80, 78, 28]. Despite valuable insights gained from these attempts, experiments are often performed on small datasets and systems are designed to merge sentences conveying *similar* information. Nonetheless, humans do not

Table 4.5: Types of sentence correspondences.

Type of PoC	Source Sentences	Summary Sentence
Pronominal Referencing	[**S1**] The bodies showed signs of torture. [**S2**] They were left on the side of a highway in Chilpancingo, about an hour north of the tourist resort of Acapulco in the state of Guerrero.	• The bodies of the men, which showed signs of torture, were left on the side of a highway in Chilpancingo.
Nominal Referencing	[**S1**] Bahamian R&B singer Johnny Kemp, best known for the 1988 party anthem "Just Got Paid," died this week in Jamaica. [**S2**] The singer is believed to have drowned at a beach in Montego Bay on Thursday, the Jamaica Constabulatory Force said in a press release.	Johnny Kemp is "believed to have drowned at a beach in Montego Bay," police say.
Common-Noun Referencing	[**S1**] A nurse confessed to killing five women and one man at hospital. [**S2**] A former nurse in the Czech Republic murdered six of her elderly patients with massive doses of potassium in order to ease her workload.	• The nurse, who has been dubbed "nurse death" locally, has admitted killing the victims with massive doses of potassium.
Repetition	[**S1**] Stewart said that she and her husband, Joseph Naaman, booked Felix on their flight from the United Arab Emirates to New York on April 1. [**S2**] The couple said they spent $1,200 to ship Felix on the 14-hour flight.	• Couple spends $1,200 to ship their cat, Felix, on a flight from the United Arab Emirates.
Event Trigger	[**S1**] Four employees of the store have been arrested, but its manager was still at large, said Goa police superintendent Kartik Kashyap. [**S2**] If convicted, they could spend up to three years in jail, Kashyap said.	• The four store workers arrested could spend 3 years each in prison if convicted.

Text cohesion can manifest itself in different forms.

restrict themselves to combine similar sentences, but also *disparate* sentences containing fundamentally different content but remain related to make fusion sensible [88]. We focus specifically on analyzing fusion of *disparate* sentences, which is a distinct problem from fusing a set of *similar* sentences.

While fusing disparate sentences is a seemingly simple task for humans to do, it has remained challenging for modern abstractive summarizers [11, 12, 46, 50]. These systems learn to perform content selection and generation through end-to-end learning. However, such a strategy is not consistently effective and they struggle to reliably perform sentence fusion [21, 25]. E.g., only 6% of summary sentences generated by pointer-generator networks [11] are fusion sentences; the ratio for human abstracts is much higher (32%). Further, Lebanoff et al. [26] report that 38% of fusion sentences contain incorrect facts. There exists a pressing need for—and this work contributes to–broadening the understanding of points of correspondence used for sentence fusion.

We present the first attempt to construct a sizeable sentence fusion dataset, where an instance in the dataset consists of a pair of input sentences, a fusion sentence, and human-annotated *points*

of correspondence between sentences. Distinguishing our work from previous efforts [65], our input contains *disparate* sentences and output is a fusion sentence containing important, though not equivalent information of the input sentences. Our investigation is inspired by Halliday and Hasan's theory of *text cohesion* [95] that covers a broad range of points of correspondence, including entity and event coreference [147, 148], shared words/concepts between sentences and more. Our contributions are as follows.

- We describe the first effort at establishing points of correspondence between disparate sentences. Without a clear understanding of points of correspondence, sentence fusion remains a daunting challenge that is only sparsely and sometimes incorrectly performed by abstractive summarizers.
- We present a sizable dataset for sentence fusion containing human-annotated corresponding regions between pairs of sentences. It can be used as a testbed for evaluating the ability of summarization models to perform sentence fusion. We report on the insights gained from annotations to suggest important future directions for sentence fusion. Our dataset is released publicly.

4.2.2 Annotating Points of Correspondence

We cast sentence fusion as a constrained summarization task where portions of text are selected from each source sentence and stitched together to form a fusion sentence; rephrasing and reordering are allowed in this process. We propose guidelines for annotating *points of correspondence* (PoC) between sentences based on Halliday and Hasan's theory of cohesion [95].

We consider points of correspondence as cohesive phrases that tie sentences together into a coherent text. Guided by text cohesion theory, we categorize PoC into five types, including pronominal referencing ("*they*"), nominal referencing ("*Johnny Kemp*"), common-noun referencing ("*five women*"), repetition, and event trigger words that are related in meaning ("*died*" and "*drowned*"). An illustration of PoC types is provided in Table 4.5. Our categorization emphasizes the lexical linking that holds a text together and gives it meaning.

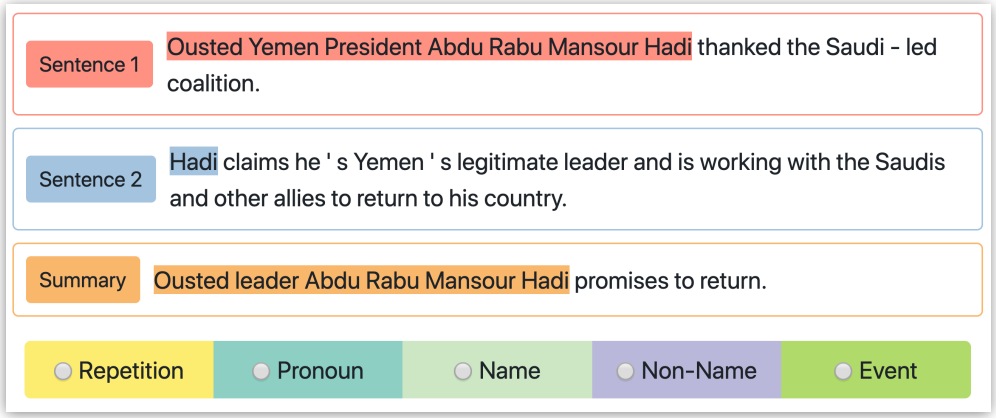

Figure 4.3: An illustration of the annotation interface. A human annotator is asked to highlight text spans referring to the same entity, then choose one from the five pre-defined PoC types.

A human annotator is instructed to identify a text span from each of the source sentences and summary sentence, thus establishing a point of correspondence between source sentences, and between source and summary sentences. As our goal is to understand the role of PoC in sentence fusion, we do not consider the case if PoC is only found in source sentences but not summary sentence, e.g., "*Kashyap said*" and "*said Goa police superintendent Kartik Kashyap*" in Table 4.5. If multiple PoC co-exist in an example, an annotator is expected to label them all; a separate PoC type will be assigned to each PoC occurrence. We are particularly interested in annotating inter-sentence PoC. If entity mentions ("*John*" and "*he*") are found in the same sentence, we do not explicitly label them but assume such intra-sentence referencing can be captured by an existing coreference resolver. Instances of source sentences and summary sentences are obtained from the test and validation splits of the CNN/DailyMail corpus [11] following the procedure described by Lebanoff et al. [26]. We take a human summary sentence as an anchor point to find two document sentences that are most similar to it based on ROUGE. It becomes an instance containing a pair of source sentences and their summary. The method allows us to identify a large quantity of candidate fusion instances.

Annotations are performed in two stages. Stage one removes all spurious pairs that are generated by the heuristic, i.e. a summary sentence that is not a valid fusion of the corresponding two source sentences. Human annotators are given a pair of sentences and a summary sentence and

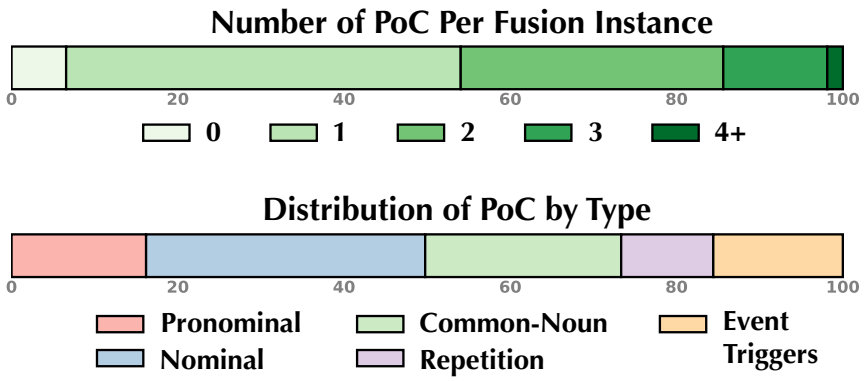

Figure 4.4: Statistics of PoC occurrences and types.

Table 4.6: Coreference resolver results.

Coref Resolver	P(%)	R(%)	F(%)	Pron.	Nominal	Comm.-Noun	Repetition	Event Trig.
SpaCy	**59.2**	20.1	30.0	30.8	23.3	10.4	39.9	2.6
AllenNLP	49.0	24.5	32.7	36.5	28.1	14.7	47.1	3.1
Stanford CoreNLP	54.2	**26.2**	**35.3**	40.0	27.3	17.4	55.1	2.3

Results of various coreference resolvers on successfully identifying inter-sentence points of correspondence (PoC) and recall scores of these resolvers split by PoC correspondence type.

are asked whether it represents a valid fusion. The pairs identified as valid fusions by a majority of annotators move on to stage two. Stage two identifies the corresponding regions in the sentences. As shown in Figure 4.3, annotators are given a pair of sentences and their summary and are tasked with highlighting the corresponding regions between each sentence. They must also choose one of the five PoC types (repetition, pronominal, nominal, common-noun referencing, and event triggers) for the set of corresponding regions.

We use Amazon mechanical turk, allowing only workers with 95% approval rate and at least 5,000 accepted tasks. To ensure high quality annotations, we first run a qualification round of 10 tasks. Workers performing sufficiently on these tasks were allowed to annotate the whole dataset. For task one, 2,200 instances were evaluated and 621 of them were filtered out. In total, we annotate points of correspondence for **1,599 instances, taken from 1,174 documents**. Similar to

[149], we report Fleiss' Kappa judged on each word (highlighted or not), yielding substantial inter-annotator agreement (κ=0.58) for annotating points of correspondence. We include a reference to the original article that each instance was taken from, thus providing context for each instance.

Figure 4.4 shows statistics of PoC occurrence frequencies and the distribution of different PoC types. A majority of sentence pairs have one or two points of correspondence. Only a small percentage (6.5%) do not share a PoC. A qualitatively analysis shows that these sentences often have an *implicit* discourse relationship, e.g., "*The two men speak. Scott then gets out of the car, again, and runs away.*" In this example, there is no clear portion of text that is shared between the sentences; rather, the connection lies in the fact that one event happens after the other. Most of the PoC are a flavor of coreference (pronominal, nominal, or common-noun). Few are exact repetition. Further, we find that only 38% of points of correspondence in the sentence pair share any words (lemmatized). This makes identifying them automatically challenging, requiring a deeper understanding of what connects the two sentences.

We contrast our dataset with previous sentence fusion datasets. McKeown et al. [87] compile a corpus of 300 sentence fusions as a first step toward a supervised fusion system. However, the input sentences have very similar meaning, though they often present lexical variations and different details. In contrast, our proposed dataset seeks to fuse significantly different meanings together into a single sentence. A large-scale dataset of sentence fusions has been recently collected [65], where each sentence has disparate content and are connected by various discourse connectives. This work instead focuses on *text cohesion* and on fusing only the salient information, which are both vital for abstractive summarization. Examples are presented in Table 4.8.

4.2.3 Resolving Coreference

Coreference resolution [147] is similar to the task of identifying points of correspondence. Thus, a natural question we ask is how well state-of-the-art coreference resolvers can be adapted to this task. If coreference resolvers can perform reasonably well on PoC identification, then these

resolvers can be used to extract PoC annotations to potentially enhance sentence fusion. If they perform poorly, coreference performance results can indicate areas of improvement for future work on detecting points of correspondence. In this work, we compare three coreference resolvers on our dataset, provided by open-source libraries: Stanford CoreNLP [150], SpaCy [151], and AllenNLP [152].

We base our evaluation on the standard metric used for coreference resolution, B-CUBED algorithm [153], with some modifications. Each resolver is run on an input pair of sentences to obtain multiple clusters, each representing an entity (e.g., *Johnny Kemp*) containing multiple mentions (e.g., *Johnny Kemp*; *he*; *the singer*) of that entity. More than one cluster can be detected by the coreference resolver, as additional entities may exist in the given sentence pair (e.g., *Johnny Kemp* and *the police*). Similarly, in Section §4.2.2, human annotators identified multiple PoC clusters, each representing a point of correspondence containing one mention from each sentence. We evaluate how well the resolver-detected clusters compare to the human-detected clusters (i.e., PoCs). If a resolver cluster overlaps both mentions for the gold-standard PoC, then this resolver cluster is classified as a hit. Any resolver cluster that does not overlap both PoC mentions is a miss. Using this metric, we can calculate precision, recall, and F1 scores based on correctly/incorrectly identified tokens from the outputs of each resolver.

The results are presented in Table 4.6. The three resolvers exhibit similar performance, but the scores on identifying points of correspondence are less than satisfying. The SpaCy resolver has the highest precision (59.2%) and Stanford CoreNLP achieves the highest F1-score (35.3%). We observe that existing coreference resolvers can sometimes struggle to use the high-level reasoning that humans use to determine what connects two sentences together. Next, we go deeper into understanding what PoC types these resolvers struggle with. We present the recall scores of these resolvers split by PoC correspondence type. Event coreference poses the most difficulty by far, which is understandable as coreference resolution only focuses on entities rather than events. More work into detecting event coreference can bring significant improvements in PoC identification. Common-noun coreference also poses a challenge, in part because names and pronouns give strong

clues as to the relationships between mentions, while common-noun relationships are more difficult to identify since they lack these clues.

4.2.4 Sentence Fusion

Truly effective summarization will only be achievable when systems have the ability to fully recognize points of correspondence between sentences. It remains to be seen whether such knowledge can be acquired implicitly by neural abstractive systems through joint content selection and generation. We next conduct an initial study to assess neural abstractive summarizers on their ability to perform sentence fusion to merge two sentences into a summary sentence. The task represents an important, atomic unit of abstractive summarization, because a long summary is still generated one sentence at a time [140].

We compare two best-performing abstractive summarizers: *Pointer-Generator* uses an encoder-decoder architecture with attention and copy mechanism [11]; *Transformer* adopts a decoder-only Transformer architecture similar to that of [48], where a summary is decoded one word at a time conditioned on source sentences and the previously-generated summary words. We use the same number of heads, layers, and units per layer as BERT-base [47]. In both cases, the summarizer was trained on about 100k instances derived from the train split of CNN/DailyMail, using the same heuristic as described in (§4.2.2) without PoC annotations. The summarizer is then tested on our dataset of 1,599 fusion instances and evaluated using standard metrics [97]. We also report how often each summarizer actually draws content from both sentences (*%Fuse*), rather than taking content from only one sentence. A generated sentence counts as a fusion if it contains at least two non-stopword tokens from each sentence not already present in the other sentence. Additionally, we include a *Concat-Baseline* creating a fusion sentence by simply concatenating the two source sentences.

The results according to the ROUGE evaluation [97] are presented in Table 4.7. Sentence fusion appears to be a challenging task even for modern abstractive summarizers. Pointer-Generator

Table 4.7: Sentence fusion results.

System	R-1	R-2	R-L	%Fuse
Concat-Baseline	36.13	18.64	27.79	99.7
Pointer-Generator	33.74	16.32	29.27	38.7
Transformer	**38.81**	**20.03**	**33.79**	50.7

ROUGE scores of neural abstractive summarizers on the sentence fusion dataset. We also report the percentage of output sentences that are indeed fusion sentences (%Fuse)

has been shown to perform strongly on abstractive summarization, but it is less so on sentence fusion and in other highly abstractive settings [43]. Transformer significantly outperforms other methods, in line with previous findings [121]. We qualitatively examine system outputs. Table 4.4 presents fusions generated by these models and exemplifies the need for infusing models with knowledge of points of correspondence. In the first example, Pointer-Generator incorrectly conflates *Robert Downey Jr.* with the *journalist* asking questions. Similarly, in the second example, Transformer states the *police officer* refused to leave when it was actually *Richards*. Had the models explicitly recognized the points of correspondence in the sentences—that *the journalist* is a separate entity from *Robert Downey Jr.* and that *Richards* is separate from *police officer*—then a more accurate summary could have been generated.

4.2.5 Conclusion

In this section, we describe a first effort at annotating points of correspondence between disparate sentences. We present a benchmark dataset comprised of the documents, source and fusion sentences, and human annotations of points of correspondence between sentences. The dataset fills a notable gap of coreference resolution and summarization research. Our findings shed light on the importance of modeling points of correspondence, suggesting important future directions for sentence fusion.

Table 4.8: Comparison of sentence fusion datasets.

[87]
[S1] Palin actually turned against the bridge project only after it became a national symbol of wasteful spending. **[S2]** Ms. Palin supported the bridge project while running for governor, and abandoned it after it became a national scandal. **[Fusion]** Palin turned against the bridge project after it became a national scandal.
DiscoFuse [65]
[S1] Melvyn Douglas originally was signed to play Sam Bailey. **[S2]** The role ultimately went to Walter Pidgeon. **[Fusion]** Melvyn Douglas originally was signed to play Sam Bailey, but the role ultimately went to Walter Pidgeon.
Points of Correspondence Dataset (Our Work)
[S1] The bodies showed signs of torture. **[S2]** They were left on the side of a highway in Chilpancingo, about an hour north of the tourist resort of Acapulco in the state of Guerrero. **[Fusion]** The bodies of the men, which showed signs of torture, were left on the side of a highway in Chilpancingo.

4.3 Learning to Fuse Sentences with Transformers using Points of Correspondence

The ability to fuse sentences is highly attractive for summarization systems because it is an essential step to produce succinct abstracts. However, to date, summarizers can fail on fusing sentences. They tend to produce few summary sentences by fusion or generate incorrect fusions that lead the summary to fail to retain the original meaning. In this section, we explore the ability of Transformers to fuse sentences and propose novel algorithms to enhance their ability to perform sentence fusion by leveraging the knowledge of *points of correspondence* between sentences. Through extensive experiments, we investigate the effects of different design choices on Transformer's performance. Our findings highlight the importance of modeling points of correspondence between sentences for effective sentence fusion. [5]

[5]This section is adapted from: L. Lebanoff, F. Dernoncourt, D. S. Kim, L. Wang, W. Chang, and F. Liu, Learning to Fuse Sentences with Transformers for Summarization, in Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020.

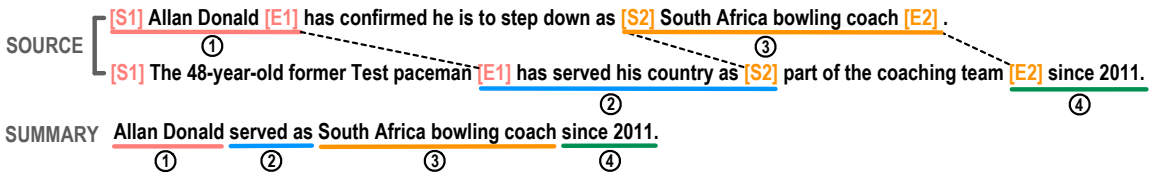

Figure 4.5: Sentence fusion involves determining what content from each sentence to retain, and how best to weave text pieces together into a well-formed sentence. Points of correspondence (PoC) are text chunks that convey the same or similar meanings, e.g., *Allan Donald* and *The 48-year-old former Test paceman*, *South Africa bowling coach* and *part of the coaching team*.

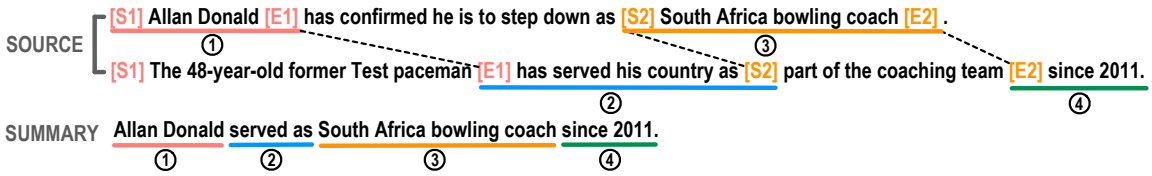

Figure 4.6: Sentence fusion involves determining what content from each sentence to retain, and how best to weave text pieces together into a well-formed sentence. Points of correspondence (PoC) are text chunks that convey the same or similar meanings, e.g., *Allan Donald* and *The 48-year-old former Test paceman*, *South Africa bowling coach* and *part of the coaching team*.

4.3.1 Introduction

A renewed emphasis must be placed on sentence fusion in the context of neural abstractive summarization. A majority of the systems are trained end-to-end [11, 10, 43, 46, 18, 50], where an abstractive summarizer is rewarded for generating summaries that contain the same words as human abstracts, measured by automatic metrics such as ROUGE [97]. A summarizer, however, is not rewarded for correctly fusing sentences. In fact, when examined more closely, only few sentences in system abstracts are generated by fusion [21, 26]. For instance, 6% of summary sentences generated by Pointer-Gen [11] are through fusion, whereas human abstracts contain 32% fusion sentences. Moreover, sentences generated by fusion are prone to errors. They can be ungrammatical, nonsensical, or otherwise ill-formed. There is thus an urgent need to develop neural abstractive summarizers to fuse sentences properly.

The importance of sentence fusion has long been recognized by the community before the era of neural text summarization. The pioneering work of Barzilay et al. [67] introduces an information fusion algorithm that combines similar elements across related text to generate a succinct summary.

Later work, such as [77, 78, 88, 28, 62], builds a dependency or word graph by combining syntactic trees of similar sentences, then employs integer linear programming to decode a summary sentence from the graph. Most of these studies have assumed a set of *similar* sentences as input, where fusion is necessary to reduce repetition. Nonetheless, humans do not limit themselves to combine similar sentences. In this section, we pay particular attention to fuse *disparate* sentences that contain fundamentally different content but remain related to make fusion sensible [88]. In Figure 4.6, we provide an example of a sentence fusion instance.

We address the challenge of fusing disparate sentences by enhancing the Transformer architecture [134] with *points of correspondence* between sentences, which are devices that tie two sentences together into a coherent text. The task of sentence fusion involves choosing content from each sentence and weaving the content pieces together into an output sentence that is linguistically plausible and semantically truthful to the original input. It is distinct from [65] that connect two sentences with discourse markers. Our contributions are as follows.

- We make crucial use of *points of correspondence* (PoC) between sentences for information fusion. Our use of PoC was initiated by the current lack of understanding of how sentences are combined in neural text summarization.
- We design new sentence fusion systems and experiment with a fusion dataset containing quality PoC annotations as the test bed for this investigation. Our findings highlight the importance of modeling points of correspondence for fusion.[6]

4.3.2 Method

A PoC is a pair of text chunks that express the same or similar meanings. In Fig. 4.6, *Allan Donald* vs. *The 48-year-old former Test paceman*, *South Africa bowling coach* vs. *part of the coaching team* are two PoCs. The use of alternative expressions for conveying the same meanings

[6]Our code is publicly available at `https://github.com/ucfnlp/sent-fusion-transformers`

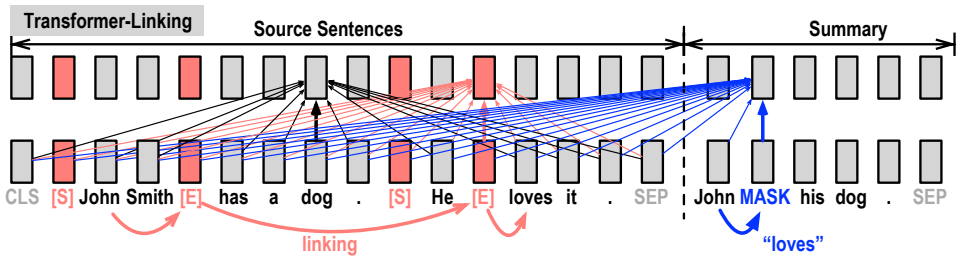

Figure 4.7: Our TRANS-LINKING model facilitates summary generation by reducing the shifting distance, allowing the model attention to shift from "John" to the tokens "[E]" then to "loves" for predicting the next summary word.

is standard practice in writing, as it increases lexical variety and reduces redundancy. However, existing summarizers cannot make effective use of these expressions to establish correspondence between sentences, often leading to ungrammatical and nonsensical outputs.

4.3.2.1 Transformer with Linking

It is advantageous for a Transformer model to make use of PoC information for sentence fusion. While Transformer-based pretrained models have had considerable success [47, 154, 14], they primarily feature pairwise relationships between *tokens*, but not PoC mentions, which are are *text chunks* of varying size. Only to a limited extent do these models embed knowledge of coreference [1], and there is a growing need for incorporating PoC linkages explicitly in a Transformer model to enhance its ability to perform sentence fusion.

We propose to enrich Transformer's source sequence with *markups* that indicate PoC linkages. Here PoC information is assumed to be available for any fusion instance (details in §4.3.3). We introduce special tokens ($[S_k]$ and $[E_k]$) to mark the start and end of each PoC mention; all mentions pertaining to the k-th PoC share the same start/end tokens. An example is illustrated in Figure 4.6, where *Allan Donald* and *The 48-year-old former Test paceman* are enriched with the same special tokens. We expect special tokens to assist in linking coreferring mentions, creating long-range dependencies between them and encouraging the model to use these mentions interchangeably in generation (Figure 4.7). The model is called "TRANS-LINKING."

Our Transformer takes as input a sequence $\mathscr{S}$ formed by concatenating the source and summary sequences. Let $\mathbf{H}^l = [\mathbf{h}_1^l, \ldots, \mathbf{h}_{|\mathscr{S}|}^l]$ be hidden representations of the l-th layer of a decoder-only architecture. An attention head transforms each vector respectively into query ($\mathbf{q}_i$), key ($\mathbf{k}_j$) and value ($\mathbf{v}_j$) vectors. The attention weight $\alpha_{i,j}$ is computed for all pairs of tokens by taking the scaled dot product of query and key vectors and applying softmax over the output (Eq. (4.1)). $\alpha_{i,j}$ indicates the importance of token j to constructing $\mathbf{h}_i^l$ of the current token i.

$$\alpha_{i,j} = \frac{\exp(\mathbf{q}_i^\top \mathbf{k}_j / \sqrt{d_k} + \mathscr{M}_{i,j})}{\sum_{j'=1}^{|\mathscr{S}|} \exp(\mathbf{q}_i^\top \mathbf{k}_{j'} / \sqrt{d_k} + \mathscr{M}_{i,j'})} \tag{4.1}$$

We utilize a mask $\mathscr{M} \in \mathbb{R}^{|\mathscr{S}| \times |\mathscr{S}|}$ to control the attention of the model (Eq. (4.2)). $\mathscr{M}_{i,j} = 0$ allows token i to attend to j and $\mathscr{M}_{i,j} = -\infty$ prevents i from attending to j as it leads $\alpha_{i,j}$ to be zero after softmax normalization. Similar to [154], a source token ($i \leq |\mathbf{x}|$) can attend to all other source tokens ($\mathscr{M}_{i,j} = 0$ for $j \leq |\mathbf{x}|$). A summary token ($i > |\mathbf{x}|$) can attend to all tokens including itself and those prior to it ($\mathscr{M}_{i,j} = 0$ for $j \leq i$). The mask $\mathscr{M}$ provides desired flexibility in terms of building hidden representations for tokens in $\mathscr{S}$. The output of the attention head is a weighted sum of the value vectors $\mathbf{h}_i^l = \sum_{j=1}^{|\mathscr{S}|} \alpha_{i,j} \mathbf{v}_j$.

$$\mathscr{M}_{i,j} = \begin{cases} 0 & \text{if } j \leq \max(i, |\mathbf{x}|) \\ -\infty & \text{otherwise} \end{cases} \tag{4.2}$$

We fine-tune the model on a sentence fusion dataset (§4.3.3) using a denoising objective, where 70% of the summary tokens are randomly masked out. The model is trained to predict the original tokens conditioned on hidden vectors of MASK tokens: $\mathbf{o} = \text{softmax}(\mathbf{W}^O \text{GeLU}(\mathbf{W}^h \mathbf{h}^L_{\text{MASK}})))$, where parameters $\mathbf{W}^O$ are tied with token embeddings. By inserting markup tokens, our model provides a soft linking mechanism to allow mentions of the same PoC to be used interchangeably in summary generation. As shown in Figure 4.7, without PoC linking, the focus of the model attention has to shift a long distance from "John" to "loves" to generate the next summary word. Their long-range dependency is not always effectively captured by the model. In contrast, our TRANS-LINKING

model substantially reduces the shifting distance, allowing the model to hop to the special token "[E]" then to "loves," facilitating summary generation.

4.3.2.2 Transformer with Shared Representation

We explore an alternative method to allow mentions of the same PoC to be connected with each other. Particularly, we direct one attention head to focus on tokens belonging to the same PoC, allowing these tokens to share semantic representations, similar to Strubell et al. [155]. Sharing representation is meaningful as these mentions are related by complex morpho-syntactic, syntactic or semantic constraints [156].

Let $\mathbf{z}$=$\{z_1, \ldots, z_{|\mathbf{z}|}\}$ be a sequence containing PoC information, where $z_i \in \{0, \ldots, \mathsf{K}\}$ indicates the index of PoC to which the token $\mathbf{x}_i$ belongs. z_i=0 indicates $\mathbf{x}_i$ is not associated with any PoC. Our TRANS-SHAREREPR model selects an attention head h from the l-th layer of the Transformer model. The attention head h governs tokens that belong to PoCs ($z_i \neq 0$). Its hidden representation $\mathbf{h}_i^l$ is computed by modeling only pairwise relationships between token i and any token j of the same PoC ($z_i = z_j$; Eq. (4.3)), while other tokens are excluded from consideration.

$$\mathscr{M}_{i,j}^{h} = \begin{cases} 0 & \text{if } i, j \leq |\mathbf{x}| \;\&\; z_i = z_j \\ -\infty & \text{otherwise} \end{cases} \tag{4.3}$$

For example, "*Allan Donald*" and "*The 48-year-old former Test paceman*" are co-referring mentions. TRANS-SHAREREPR allows these tokens to only attend to each other when learning representations using the attention head h. These tokens are likely to yield similar representations. The method thus accomplishes a similar goal as TRANS-LINKING to allow *tokens* of the same PoC to be treated equivalently during summary generation; we explore the selection of attention heads in §4.3.3.

Table 4.9: Results of various sentence fusion systems.

System	Heuristic Set				Point of Correspondence Test Set						
	R-1	**R-2**	**R-L**	**BLEU**	**R-1**	**R-2**	**R-L**	**BLEU**	**B-Score**	**#Tkns**	**%Fuse**
Pointer-Generator	35.8	18.2	31.8	41.9	33.7	16.3	29.3	40.3	57.3	14.3	38.7
Transformer	39.6	20.9	35.3	47.2	38.8	20.0	33.8	45.8	61.3	15.1	50.7
Trans-LINKING	**39.8**	**21.1**	**35.3**	**47.3**	38.8	20.1	33.9	45.5	61.1	15.1	**55.8**
Trans-SHAREREPR	39.4	20.9	35.2	46.9	**39.0**	**20.2**	**33.9**	**45.8**	61.2	**15.2**	46.5
Concat-Baseline	37.2	20.0	28.7	25.0	36.1	18.6	27.8	24.6	60.4	52.0	99.7

We report the percentage of output sentences that are generated by fusion (%Fuse) and the average number of tokens per output sentence (#Tkns). To calculate %Fuse, we follow the same procedure used by [66] – a generated sentence is regarded as a fusion if it contains at least two non-stopword tokens from each sentence that do not already exist in the other sentence.

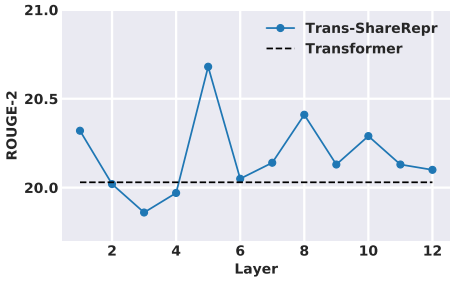

Figure 4.8: The first attention head from the l-th layer is dedicated to coreferring mentions. The head encourages tokens of the same PoC to share similar representations. Our results suggest that the attention head of the 5-th layer achieves competitive performance, while most heads perform better than the baseline. The findings are congruent with [1] that provides a detailed analysis of BERT's attention.

4.3.3 Experiments

Corpus Our corpus contains a collection of documents, source and fusion sentences, and human annotations of corresponding regions between sentences. The set of documents were sampled from CNN/DM [11] and PoC annotations were obtained from Lebanoff et al. [66]. They use a human summary sentence as an anchor point to find two document sentences that are most similar to it, which forms a fusion instance containing a pair of source sentences and their summary. PoCs have been annotated based on Halliday and Hasan's theory of cohesion [95] for 1,494 fusion instances, taken from 1,174 documents in the test and valid splits of CNN/DM with a moderate to high inter-annotator agreement (0.58).

Automatic Evaluation We proceed by investigating the effectiveness of various sentence fusion models, including (a) Pointer-Generator [11] that employs an encoder-decoder architecture to condense input sentences to a vector representation, then decode it into a fusion sentence. (b) Transformer, our baseline Transformer architecture w/o PoC information. It is a strong baseline that resembles the UniLM model described in [154]. (c) Trans-Linking uses special tokens to mark the boundaries of PoC mentions (§4.3.2.1). (d) Trans-ShareRepr allows tokens of the same PoC to share representations (§4.3.2.2). All Transformer models are initialized with BERT-BASE parameters and are fine-tuned using UniLM's sequence-to-sequence objective for 11 epochs, with a batch size of 32. The source and fusion sentences use BPE tokenization, and the combined input/output sequence is truncated to 128 tokens. We use the Adam optimizer with a learning rate of 2e-5 with warm-up. For PG, we use the default settings and truncate the output sequences to 60 tokens.

All of the fusion models are trained (or fine-tuned) on the same training set containing 107k fusion instances from the training split of CNN/DM; PoC are identified by the spaCy coreference resolver. We evaluate fusion models on two test sets, including a "heuristic set" containing testing instances and automatically identified PoC via spaCy, and a final test set containing 1,494 instances with human-labelled PoC. We evaluate only on the instances that contain at least one point of

correspondence, so we have to disregard a small percentage of instances (6.6%) in the dataset of Lebanoff et al. [66] that contain no points of correspondence.

Table 4.10: Example output of sentence fusion systems.

Source: Later that month, the ICC opened a preliminary examination into the situation in Palestinian territories, paving the way for possible war crimes investigations against Israelis. Israel and the United States, neither of which is an ICC member, opposed the Palestinians' efforts to join the body.
Pointer-Generator: *ICC opened a preliminary examination into the situation in Palestinian territories .*
Transformer: *Israel, U.S. and the United States are investigating possible war crimes, paving way for war crimes.*
Transformer-ShareRepr: *Israel and U.S. opposed the ICC's investigation into the situation in Palestinian territories.*
Reference: *Israel and the United States opposed the move, which could open the door to war crimes investigations against Israelis.*

PG only performs sentence shortening rather than fusion. **Transformer** fails to retain the original meaning and **Transformer-ShareRepr** performs best. **Reference** demonstrates a high level of abstraction. Sentences are manually de-tokenized for readability.

We compare system outputs and references using a number of automatic evaluation metrics including ROUGE [97], BLEU [157] and BERTScore [92]. Results are presented in Table 4.9. We observe that all Transformer models outperform PG, suggesting that these models can benefit substantially from unsupervised pretraining on a large corpus of text. On the heuristic test set where training and testing conditions match (they both use automatically identified PoC), **Trans-Linking** performs better than **Trans-ShareRepr**, and vice versa on the final test set. We conjecture that this is because the linking model has a stronger requirement on PoC boundaries and the training/testing conditions must match for it to be effective. In contrast, **Trans-ShareRepr** is more lenient with mismatched conditions.

We include a **Concat-Baseline** that creates a fusion by simply concatenating two input sentences. Its output contains 52 tokens on average, while other model outputs contain 15 tokens. This is a 70% compression rate, which adds to the challenge of content selection [111]. Despite that all models are trained to fuse sentences, their outputs are not guaranteed to be fusions and shortening of single

Table 4.11: Human and extractiveness evaluation.

System	Truthful.	Extractiveness 1-gram	2-gram	3-gram
Pointer-Generator	63.6	97.5	83.1	72.8
Transformer	71.7	91.9	68.6	54.2
Trans-SHAREREPR	70.9	92.0	70.1	56.4
Reference	67.2	72.0	34.9	20.9

Fusion sentences are evaluated by their level of truthfulness and extractivenss. Our system fusions attain a high level of truthfulness with moderate extractivenss.

sentences is possible. We observe that Trans-Linking has the highest rate of producing fusions (56%). In Figure 4.8, we examine the effect of different design choices, where the first attention head of the l-th layer is dedicated to PoC. We report the averaged results in Table 4.9.

Human evaluation We investigate the quality of fusions with human evaluation. The models we use for comparison include (a) Pointer-Generator, (b) Transformer, (c) Trans-ShareRepr and (d) human reference fusion sentences. Example outputs for each model can be seen in Table 4.10. We perform evaluation on 200 randomly sampled instances from the point of correspondence test set. We take an extra step to ensure all model outputs for selected instances contain fusion sentences, as opposed to shortening of single sentences. A human evaluator from Amazon Mechanical Turk (`mturk.com`) is asked to assess if the fusion sentence has successfully retained the original meaning. Specifically, an evaluator is tasked with reading the two article sentences and fusion sentence and answering yes or no to the following question, "Is this summary sentence true to the original article sentences it's been sourced from, and it has not added any new meaning?" Each instance is judged by five human evaluators and results are shown in Table 4.11. Additionally, we measure their *extractiveness* by reporting on the percentage of n-grams (n=1/2/3) that appear in the source. Human sentence fusions are highly abstractive, and as the gold standard, we wish to emulate this level of abstraction in automatic summarizers. Fusing two sentences together coherently requires connective phrases and sometimes requires rephrasing parts of sentences. However, higher abstraction does not mean higher quality fusions, especially in neural models.

Interestingly, we observe that humans do not always rate reference fusions as truthful. This is in part because reference fusions exhibit a high level of abstraction and they occasionally contain content not in the source. If fusion sentences are less extractive, humans sometimes perceive that as less truthful, especially when compared to fusions that reuse the source text. Our results call for a reexamination of sentence fusion using better evaluation metrics including semantics and question-answering-based metrics [91, 94, 93].

4.3.4 Conclusion

We address the challenge of information fusion in the context of neural abstractive summarization by making crucial use of points of correspondence between sentences. We enrich Transformers with PoC information and report model performance on a new test bed for information fusion. Our findings suggest that modeling points of correspondence is crucial for effective sentence fusion, and sentence fusion remains a challenging direction of research. Future work may explore the use of points of correspondence and sentence fusion in the standard setting of document summarization. Performing sentence fusion accurately and succinctly is especially important for summarizing long documents and book chapters [158]. These domains may contain more entities and events to potentially confuse a summarizer, making our method of explicitly marking these entities beneficial.

CHAPTER 5: MULTI-DOCUMENT SUMMARIZATION

Multi-document summarization is challenging. A summarizer must be able to read and understand multiple input documents about one topic and condense it into a short summary. It must also be able to recognize redundant information between documents. Separating the task into two explicit steps of content selection and surface realization can make this task easier.

In Section 5.1, we present an approach for adapting an encoder-decoder model trained on single documents to the multi-document setting. Single-document datasets have much greater amounts of data, allowing an encoder-decoder model to be trained effectively and act as a surface realization model. We can then use a separate algorithm that does not require any training data – Maximal Marginal Relevance (MMR) – to perform content selection.

In Section 5.2, we introduce a method based on *endorsement* between documents. A segment of text can be endorsed by another document if that document also contains a similar segment of text. Segments of text that receive endorsement from many other documents are included in the final summary. To incorporate this endorsement information into a surface realization model, we introduce companion heads to the Transformer architecture. This method shows improvement over baselines on three multi-document summarization datasets.

5.1 Adapting the Encoder-Decoder Model from Single-Document to Multi-Document Summarization

Generating a text abstract from a set of documents remains a challenging task. The neural encoder-decoder framework has recently been exploited to summarize single documents, but its success can in part be attributed to the availability of large parallel data automatically acquired from the Web. In contrast, parallel data for multi-document summarization are scarce and costly to obtain. There is a pressing need to adapt an encoder-decoder model trained on single-document summarization data to work with multiple-document input. In this section, we present an initial investigation into a novel adaptation method. It exploits the maximal marginal relevance method to select representative sentences from multi-document input, and leverages an abstractive encoder-decoder model to fuse disparate sentences to an abstractive summary. The adaptation method is robust and itself requires no training data. Our system compares favorably to state-of-the-art extractive and abstractive approaches judged by automatic metrics and human assessors. [1]

5.1.1 Introduction

Neural abstractive summarization has primarily focused on summarizing short texts written by single authors. For example, *sentence summarization* seeks to reduce the first sentence of a news article to a title-like summary [7, 8, 159, 55]; *single-document summarization* (**SDS**) focuses on condensing a news article to a handful of bullet points [10, 11]. These summarization studies are empowered by large parallel datasets automatically harvested from online news outlets, including Gigaword [7], CNN/Daily Mail [114], NYT [160], and Newsroom [115].

To date, *multi-document summarization* (**MDS**) has not yet fully benefited from the development of neural encoder-decoder models. MDS seeks to condense a set of documents likely written

[1] This section is adapted from: L. Lebanoff, K. Song, and F. Liu, Adapting the Neural Encoder-Decoder Framework from Single to Multi-Document Summarization, in Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2018.

by multiple authors to a short and informative summary. It has practical applications, such as summarizing product reviews [70], student responses to post-class questionnaires [161, 162], and sets of news articles discussing certain topics [163]. State-of-the-art MDS systems are mostly extractive [101]. Despite their promising results, such systems cannot perform text abstraction, e.g., paraphrasing, generalization, and sentence fusion [164]. Further, annotated MDS datasets are often scarce, containing only hundreds of training pairs (see Table 2.1). The cost to create ground-truth summaries from multiple-document inputs can be prohibitive. The MDS datasets are thus too small to be used to train neural encoder-decoder models with millions of parameters without overfitting.

A promising route to generating an abstractive summary from a multi-document input is to apply a neural encoder-decoder model trained for single-document summarization to a "*mega-document*" created by concatenating all documents in the set at test time. Nonetheless, such a model may not scale well for two reasons. First, identifying important text pieces from a mega-document can be challenging for the encoder-decoder model, which is trained on single-document summarization data where the summary-worthy content is often contained in the first few sentences of an article. This is not the case for a mega-document. Second, redundant text pieces in a mega-document can be repeatedly used for summary generation under the current framework. The attention mechanism of an encoder-decoder model [165] is position-based and lacks an awareness of semantics. If a text piece has been attended to during summary generation, it is unlikely to be used again. However, the attention value assigned to a similar text piece in a different position is not affected. The same content can thus be repeatedly used for summary generation. These issues may be alleviated by improving the encoder-decoder architecture and its attention mechanism [33, 9]. However, in these cases the model has to be re-trained on large-scale MDS datasets that are not available at the current stage. There is thus an increasing need for a lightweight adaptation of an encoder-decoder model trained on SDS datasets to work with multi-document inputs at test time.

In this section, we present a novel adaptation method, named PG-MMR, to generate abstracts from multi-document inputs. The method is robust and requires no MDS training data. It com-

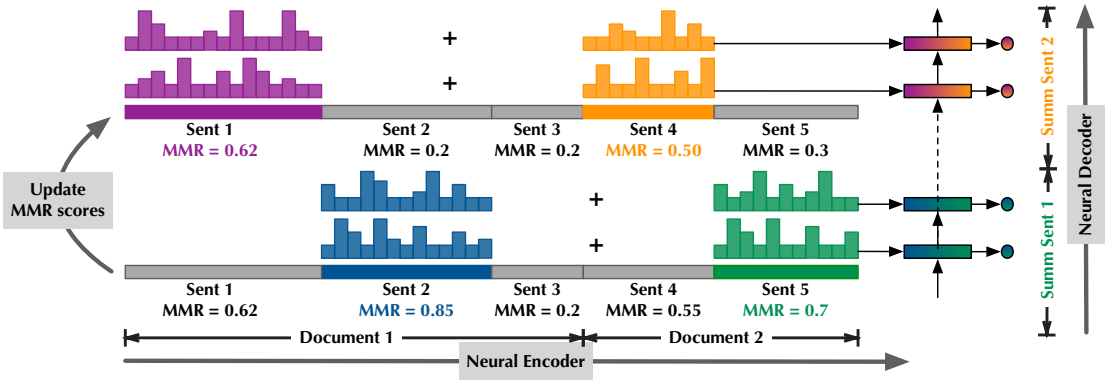

Figure 5.1: System framework. The PG-MMR system uses K highest-scored source sentences (in this case, K=2) to guide the PG model to generate a summary sentence. All other source sentences are "muted" in this process. Best viewed in color.

bines a recent neural encoder-decoder model (PG for Pointer-Generator networks; See et al., 2017) that generates abstractive summaries from single-document inputs with a strong extractive summarization algorithm (MMR for Maximal Marginal Relevance; Carbonell and Goldstein, 1998) that identifies important source sentences from multi-document inputs. The PG-MMR algorithm iteratively performs the following. It identifies a handful of the most important sentences from the mega-document. The attention weights of the PG model are directly modified to focus on these important sentences when generating a summary sentence. Next, the system re-identifies a number of important sentences, but the likelihood of choosing certain sentences is reduced based on their similarity to the partially-generated summary, thereby reducing redundancy. Our research contributions include the following:

- we present an investigation into a novel adaptation method of the encoder-decoder framework from single- to multi-document summarization. To the best of our knowledge, this is the first attempt to couple the maximal marginal relevance algorithm with pointer-generator networks for multi-document summarization;
- we demonstrate the effectiveness of the proposed method through extensive experiments on standard MDS datasets. Our system compares favorably to state-of-the-art extractive and abstractive summarization systems measured by both automatic metrics and human judgments.

5.1.2 Limits of the Encoder-Decoder Model

The encoder-decoder architecture has become the *de facto* standard for neural abstractive summarization [7]. The encoder is often a bidirectional LSTM [166] converting the input text to a set of hidden states $\{\mathbf{h}_i^e\}$, one for each input word, indexed by i. The decoder is a unidirectional LSTM that generates a summary by predicting one word at a time. The decoder hidden states are represented by $\{\mathbf{h}_t^d\}$, indexed by t. For sentence and single-document summarization [8, 10, 11], the input text is treated as a sequence of words, and the model is expected to capture the source syntax inherently.

$$e_{t,i} = \mathbf{v}^\top \tanh(\mathbf{W}^e[\mathbf{h}_t^d || \mathbf{h}_i^e || \widetilde{\alpha}_{t,i}] + \mathbf{b}^e) \tag{5.1}$$

$$\alpha_{t,i} = \text{softmax}(e_{t,i}) \tag{5.2}$$

$$\widetilde{\alpha}_{t,i} = \textstyle\sum_{t'=0}^{t-1} \alpha_{t',i} \tag{5.3}$$

The attention weight $\alpha_{t,i}$ measures how important the i-th input word is to generating the t-th output word (Eq. (5.1-5.2)). Following [11], $\alpha_{t,i}$ is calculated by measuring the strength of interaction between the decoder hidden state $\mathbf{h}_t^d$, the encoder hidden state $\mathbf{h}_i^e$, and the *cumulative* attention $\widetilde{\alpha}_{t,i}$ (Eq. (5.3)). $\widetilde{\alpha}_{t,i}$ denotes the cumulative attention that the i-th input word receives up to time step t-1. A large value of $\widetilde{\alpha}_{t,i}$ indicates the i-th input word has been used prior to time t and it is unlikely to be used again for generating the t-th output word.

A context vector ($\mathbf{c}_t$) is constructed (Eq. (5.4)) to summarize the semantic meaning of the input; it is a weighted sum of the encoder hidden states. The context vector and the decoder hidden state ($[\mathbf{h}_t^d || \mathbf{c}_t]$) are then used to compute the vocabulary probability $P_{vcb}(w)$ measuring the likelihood of

a vocabulary word w being selected as the t-th output word (Eq. (5.5)).[2]

$$\mathbf{c}_t = \sum_i \alpha_{t,i} \mathbf{h}_i^e \tag{5.4}$$

$$P_{vcb}(w) = \text{softmax}(\mathbf{W}^y[\mathbf{h}_t^d || \mathbf{c}_t] + \mathbf{b}^y) \tag{5.5}$$

In many encoder-decoder models, a "switch" is estimated ($p_{gen} \in [0,1]$) to indicate whether the system has chosen to select a word from the vocabulary or to copy a word from the input text (Eq. (5.6)). The switch is computed using a feedforward layer with σ activation over $[\mathbf{h}_t^d || \mathbf{c}_t || \mathbf{y}_{t-1}]$, where $\mathbf{y}_{t-1}$ is the embedding of the output word at time t-1. The attention weights ($\alpha_{t,i}$) are used to compute the copy probability (Eq. (5.7)). If a word w appears once or more in the input text, its copy probability ($\sum_{i:w_i=w} \alpha_{t,i}$) is the sum of the attention weights over all its occurrences. The final probability $P(w)$ is a weighted combination of the vocabulary probability and the copy probability. A cross-entropy loss function can often be used to train the model end-to-end.

$$p_{gen} = \sigma(\mathbf{w}^z[\mathbf{h}_t^d || \mathbf{c}_t || \mathbf{y}_{t-1}]) + b^z) \tag{5.6}$$

$$P(w) = p_{gen} P_{vcb}(w) + (1 - p_{gen}) \sum_{i:w_i=w} \alpha_{t,i} \tag{5.7}$$

To thoroughly understand the aforementioned encoder-decoder model, we divide its model parameters into four groups. They include

- parameters of the encoder and the decoder;
- $\{\mathbf{w}^z, b^z\}$ for calculating the "switch" (Eq. (5.6));
- $\{\mathbf{W}^y, \mathbf{b}^y\}$ for calculating $P_{vcb}(w)$ (Eq. (5.5));
- $\{\mathbf{v}, \mathbf{W}^e, \mathbf{b}^e\}$ for attention weights (Eq. (5.1)).

[2]Here $[\cdot||\cdot]$ represents the concatenation of two vectors. The pointer-generator networks [11] use two linear layers to produce the vocabulary distribution $P_{vcb}(w)$. We use $\mathbf{W}^y$ and $\mathbf{b}^y$ to denote parameters of both layers.

By training the encoder-decoder model on single-document summarization (SDS) data containing a large collection of news articles paired with summaries [114], these model parameters can be effectively learned.

However, at test time, we wish for the model to generate abstractive summaries from *multi-document inputs*. This brings up two issues. First, the parameters are ineffective at identifying salient content from multi-document inputs. Humans are very good at identifying representative sentences from a set of documents and fusing them into an abstract. However, this capability is not supported by the encoder-decoder model. Second, the attention mechanism is based on input word positions but not their semantics. It can lead to redundant content in the multi-document input being repeatedly used for summary generation. We conjecture that both aspects can be addressed by introducing an "external" model that selects representative sentences from multi-document inputs and dynamically adjusts the sentence importance to reduce summary redundancy. This external model is integrated with the encoder-decoder model to generate abstractive summaries using selected representative sentences. In the following section we present our adaptation method for multi-document summarization.

5.1.3 Our Method

Maximal marginal relevance. Our adaptation method incorporates the maximal marginal relevance algorithm (MMR; Carbonell and Goldstein, 1998) into pointer-generator networks (PG; See et al., 2017) by adjusting the network's attention values. MMR is one of the most successful extractive approaches and, despite its straightforwardness, performs on-par with state-of-the-art systems [161, 30]. At each iteration, MMR selects one sentence from the document (D) and includes it in the summary (S) until a length threshold is reached. The selected sentence (s_i) is the most important one amongst the remaining sentences and it has the least content overlap with the current summary. In the equation below, $\text{Sim}_1(s_i, D)$ measures the similarity of the sentence s_i to the document. It serves as a proxy of sentence importance, since important sentences usually show

similarity to the centroid of the document. $\max_{s_j \in S} \mathrm{Sim}_2(s_i, s_j)$ measures the maximum similarity of the sentence s_i to each of the summary sentences, acting as a proxy of redundancy. λ is a balancing factor.

$$\underset{s_i \in D \setminus S}{\arg\max} \Big[\underbrace{\lambda \mathrm{Sim}_1(s_i, D)}_{\text{importance}} - \underbrace{(1-\lambda) \max_{s_j \in S} \mathrm{Sim}_2(s_i, s_j)}_{\text{redundancy}} \Big] \tag{5.8}$$

Our PG-MMR describes an iterative framework for summarizing a multi-document input to a summary consisting of multiple sentences. At each iteration, PG-MMR follows the MMR principle to select the K highest-scored source sentences; they serve as the basis for PG to generate a summary sentence. After that, the scores of all source sentences are updated based on their importance and redundancy. Sentences that are highly similar to the partial summary receive lower scores. Selecting K sentences via the MMR algorithm helps the PG system to effectively identify salient source content that has not been included in the summary.

Muting. To allow the PG system to effectively utilize the K source sentences without retraining the neural model, we dynamically adjust the PG attention weights ($\alpha_{t,i}$) at test time. Let S_k represent a selected sentence. The attention weights of the words belonging to $\{\mathrm{S}_k\}_{k=1}^{K}$ are calculated as before. However, words in other sentences are forced to receive zero attention weights ($\alpha_{t,i}$=0), and all $\alpha_{t,i}$ are renormalized (Eq. (5.9)).

$$\alpha_{t,i}^{\text{new}} = \begin{cases} \alpha_{t,i} & i \in \{\mathrm{S}_k\}_{k=1}^{K} \\ 0 & \text{otherwise} \end{cases} \tag{5.9}$$

It means that the remaining sentences are "muted" in this process. In this variant, the sentence importance does not affect the original attention weights, other than muting.

In an alternative setting, the sentence salience is multiplied with the word salience and renormalized (Eq. (5.10)). PG uses the reweighted alpha values to predict the next summary word.

$$\alpha_{t,i}^{\text{new}} = \begin{cases} \alpha_{t,i}\text{MMR}(\text{S}_k) & i \in \{\text{S}_k\}_{k=1}^{K} \\ 0 & \text{otherwise} \end{cases} \tag{5.10}$$

Sentence Importance. To estimate sentence importance $\text{Sim}_1(s_i, D)$, we introduce a supervised regression model in this work. Importantly, the model is trained on single-document summarization datasets where training data are abundant. At test time, the model can be applied to identify important sentences from multi-document input. Our model determines sentence importance based on four indicators, inspired by how humans identify important sentences from a document set. They include (a) sentence length, (b) its absolute and relative position in the document, (c) sentence quality, and (d) how close the sentence is to the main topic of the document set. These features are considered to be important indicators in previous extractive summarization framework [5, 163].

Regarding the sentence quality (c), we leverage the PG model to build the sentence representation. We use the bidirectional LSTM encoder to encode any source sentence to a vector representation. $[\overrightarrow{\mathbf{h}_N^e}||\overleftarrow{\mathbf{h}_1^e}]$ is the concatenation of the last hidden states of the forward and backward passes. A document vector is the average of all sentence vectors. We use the document vector and the cosine similarity between the document and sentence vectors as indicator (d). A support vector regression model is trained on (sentence, score) pairs where the training data are obtained from the CNN/Daily Mail dataset. The target importance score is the ROUGE-L recall of the sentence compared to the ground-truth summary. Our model architecture leverages neural representations of sentences and documents, they are data-driven and not restricted to a particular domain.

Sentence Redundancy. To calculate the redundancy of the sentence ($\max_{s_j \in S} \text{Sim}_2(s_i, s_j)$), we compute the ROUGE-L precision, which measures the longest common subsequence between a source sentence and the partial summary (consisting of all sentences generated thus far by the PG model), divided by the length of the source sentence. A source sentence yielding a high ROUGE-L

precision is deemed to have significant content overlap with the partial summary. It will receive a low MMR score and hence is less likely to serve as basis for generating future summary sentences.

Alg. 1 provides an overview the PG-MMR algorithm and Fig. 5.3 is a graphical illustration. The MMR scores of source sentences are updated after each summary sentence is generated by the PG model. Next, a different set of highest-scored sentences are used to guide the PG model to generate the next summary sentence. "Muting" the remaining source sentences is important because it helps the PG model to focus its attention on the most significant source content. The code for our model is publicly available to further MDS research.[3]

Algorithm 1 The PG-MMR algorithm for summarizing multi-document inputs.

Input: SDS data; MDS source sentences $\{S_i\}$

Train the PG model on SDS data
▶ $I(S_i)$ and $R(S_i)$ are the importance and redundancy scores of the source sentence S_i
$I(S_i) \leftarrow \text{SVR}(S_i)$ for all source sentences
$\text{MMR}(S_i) \leftarrow \lambda I(S_i)$ for all source sentences
Summary $\leftarrow \{\}$
$t \leftarrow$ index of summary words
while $t < L_{\max}$ **do**
  Find $\{S_k\}_{k=1}^{K}$ with highest MMR scores
  Compute $\alpha_{t,i}^{\text{new}}$ based on $\{S_k\}_{k=1}^{K}$ (Eq. (5.9))
  Run PG decoder for one step to get $\{w_t\}$
  Summary $\leftarrow$ Summary + $\{w_t\}$
  if w_t is the period symbol **then**
    $R(S_i) \leftarrow \text{Sim}(S_i, \text{Summary}), \forall i$
    $\text{MMR}(S_i) \leftarrow (S_i) - (1-\lambda) R(S_i), \forall i$
  end if
end while

5.1.4 Experimental Setup

Datasets. We investigate the effectiveness of the PG-MMR method by testing it on standard multi-document summarization datasets [112, 113]. These include DUC-03, DUC-04, TAC-08, TAC-

[3] https://github.com/ucfnlp/multidoc_summarization

10, and TAC-11, containing 30/50/48/46/44 topics respectively. The summarization system is tasked with generating a concise, fluent summary of 100 words or less from a set of 10 documents discussing a topic. All documents in a set are chronologically ordered and concatenated to form a mega-document serving as input to the PG-MMR system. Sentences that start with a quotation mark or do not end with a period are excluded [52]. Each system summary is compared against 4 human abstracts created by NIST assessors. Following convention, we report results on DUC-04 and TAC-11 datasets, which are standard test sets; DUC-03 and TAC-08/10 are used as a validation set for hyperparameter tuning.[4]

The PG model is trained for single-document summarization using the CNN/Daily Mail [114] dataset, containing single news articles paired with summaries (human-written article highlights). The training set contains 287,226 articles. An article contains 781 tokens on average; and a summary contains 56 tokens (3.75 sentences). During training we use the hyperparameters provided by See et al. [11]. At test time, the maximum/minimum decoding steps are set to 120/100 words respectively, corresponding to the max/min lengths of the PG-MMR summaries. Because the focus of this work is on multi-document summarization (MDS), we do not report results for the CNN/Daily Mail dataset.

Baselines. We compare PG-MMR against a broad spectrum of baselines, including state-of-the-art extractive ('*ext*-') and abstractive ('*abs*-') systems. They are described below.[5]

- *ext*-**SumBasic** [138] is an extractive approach assuming words occurring frequently in a document set are more likely to be included in the summary;
- *ext*-**KL-Sum** [102] greedily adds source sentences to the summary if it leads to a decrease in KL divergence;
- *ext*-**LexRank** [139] uses a graph-based approach to compute sentence importance based on eigenvector centrality in a graph representation;
- *ext*-**Centroid** [163] computes the importance of each source sentence based on its cosine similarity with the document centroid;

[4]The hyperparameters for all PG-MMR variants are K=7 and λ=0.6; except for "w/ BestSummRec" where K=2.

[5]We are grateful to Hong et al. [163] for providing the summaries generated by Centroid, ICSISumm, DPP systems. These are only available for the DUC-04 dataset.

Table 5.1: ROUGE results on the DUC-04 dataset.

System	DUC-04 R-1	R-2	R-SU4
SumBasic [138]	29.48	4.25	8.64
KLSumm (Haghighi et al., 2009)	31.04	6.03	10.23
LexRank [139]	34.44	7.11	11.19
Centroid [163]	35.49	7.80	12.02
ICSISumm [4]	37.31	9.36	13.12
DPP [103]	38.78	9.47	13.36
Extract+Rewrite [55]	28.90	5.33	8.76
Opinosis [69]	27.07	5.03	8.63
PG-Original [11]	31.43	6.03	10.01
PG-MMR w/ SummRec	34.57	7.46	11.36
PG-MMR w/ SentAttn	36.52	8.52	12.57
PG-MMR w/ Cosine (*default*)	36.88	**8.73**	12.64
PG-MMR w/ BestSummRec	36.42	**9.36**	13.23

Table 5.2: ROUGE results on the TAC-11 dataset

System	TAC-11 R-1	R-2	R-SU4
SumBasic [138]	31.58	6.06	10.06
KLSumm (Haghighi et al., 2009)	31.23	7.07	10.56
LexRank [139]	33.10	7.50	11.13
Extract+Rewrite [55]	29.07	6.11	9.20
Opinosis [69]	25.15	5.12	8.12
PG-Original [11]	31.44	6.40	10.20
PG-MMR w/ SummRec	35.06	8.72	12.39
PG-MMR w/ SentAttn	37.01	10.43	13.85
PG-MMR w/ Cosine (*default*)	37.17	**10.92**	14.04
PG-MMR w/ BestSummRec	40.44	**14.93**	17.61

- *ext*-**ICSISumm** [167] leverages the ILP framework to identify a globally-optimal set of sentences covering the most important concepts in the document set;
- *ext*-**DPP** [103] selects an optimal set of sentences per the determinantal point processes that balance the coverage of important information and the sentence diversity;
- *abs*-**Opinosis** [69] generates abstractive summaries by searching for salient paths on a word co-occurrence graph created from source documents;
- *abs*-**Extract+Rewrite** [55] is a recent approach that scores sentences using LexRank and generates a title-like summary for each sentence using an encoder-decoder model trained on Gigaword data.
- *abs*-**PG-Original** [11] introduces an encoder-decoder model that encourages the system to copy words from the source text via pointing, while retaining the ability to produce novel words through the generator.

5.1.5 Results

Having described the experimental setup, we next compare the PG-MMR method against the baselines on standard MDS datasets, evaluated by both automatic metrics and human assessors.

ROUGE [97]. This automatic metric measures the overlap of unigrams (R-1), bigrams (R-2) and skip bigrams with a maximum distance of 4 words (R-SU4) between the system summary and a set of reference summaries. ROUGE scores of various systems are presented in Table 5.9 and 5.10 respectively for the DUC-04 and TAC-11 datasets.

We explore variants of the PG-MMR method. They differ in how the importances of source sentences are estimated and how the sentence importance affects word attention weights. "***w/ Cosine***" computes the sentence importance as the cosine similarity score between the sentence and document vectors, both represented as sparse TF-IDF vectors under the vector space model. "***w/ SummRec***" estimates the sentence importance as the predicted R-L recall score between the sentence and the summary. A support vector regression model is trained on sentences from the CNN/Daily Mail datasets (≈33K) and applied to DUC/TAC sentences at test time (see §5.1.3). "***w/ BestSummRec***" obtains the best estimate of sentence importance by calculating the R-L recall score between the sentence and reference summaries. It serves as an upper bound for the performance of "w/ SummRec." For all variants, the sentence importance scores are normalized to the range of [0,1]. "***w/ SentAttn***" adjusts the attention weights so that words in important sentences are more likely to be used to generate the summary. The weights are otherwise computed using Eq. (5.9).

As seen in Table 5.9 and 5.10, our PG-MMR method surpasses all unsupervised extractive baselines, including SumBasic, KLSumm, and LexRank. On the DUC-04 dataset, ICSISumm and DPP show good performance, but these systems are trained directly on MDS datasets, which are not utilized by the PG-MMR method. PG-MMR exhibits superior performance compared to existing abstractive systems. It outperforms Opinosis and PG-Original by a large margin in terms of R-2 F-scores (5.03/6.03/**8.73** for DUC-04 and 5.12/6.40/**10.92** for TAC-11). In particular,

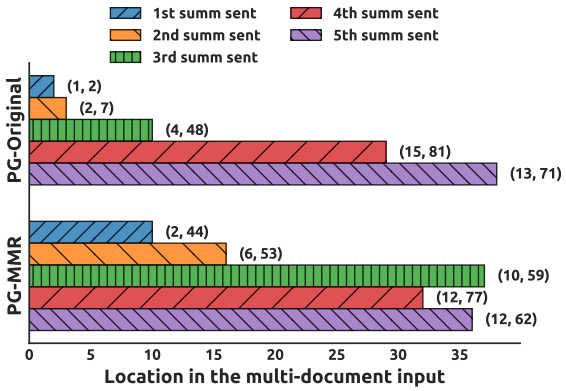

Figure 5.2: The median location of summary n-grams in the multi-document input (and the lower/higher quartiles). The n-grams come from the 1st/2nd/3rd/4th/5th summary sentence and the location is the source sentence index. (TAC-11)

Table 5.3: Extractiveness results.

System	1-grams	2-grams	3-grams	Sent
Extr+Rewrite	89.37	54.34	25.10	6.65
PG-Original	99.64	96.28	88.83	47.67
PG-MMR	99.74	97.64	91.57	59.13
Human Abst.	84.32	45.22	18.70	0.23

Percentages of summary n-grams (or the entire sentences) appear in the multi-document input. (TAC-11)

PG-Original is the original pointer-generator networks with multi-document inputs at test time. Compared to it, PG-MMR is more effective at identifying summary-worthy content from the input. "w/ Cosine" is used as the default PG-MMR and it shows better results than "w/ SummRec." It suggests that the sentence and document representations obtained from the encoder-decoder model (trained on CNN/DM) are suboptimal, possibly due to a vocabulary mismatch, where certain words in the DUC/TAC datasets do not appear in CNN/DM and their embeddings are thus not learned during training. Finally, we observe that "w/ BestSummRec" yields the highest performance on both datasets. This finding suggests that there is a great potential for improvements of the PG-MMR method as its "extractive" and "abstractive" components can be separately optimized.

Location of summary content. We are interested in understanding why PG-MMR outperforms PG-Original at identifying summary content from the multi-document input. We ask the question: where, in the source documents, does each system tend to look when generating their summaries?

Table 5.4: Linguistic quality and rankings of system summaries. (DUC-04)

System	Linguistic Quality			Rankings (%)			
	Fluency	Inform.	NonRed.	1st	2nd	3rd	4th
Extract+Rewrite	2.03	2.19	1.88	5.6	11.6	11.6	**71.2**
LexRank	**3.29**	3.36	3.30	30.0	28.8	32.0	9.2
PG-Original	3.20	3.30	3.19	29.6	26.8	**32.8**	10.8
PG-MMR	3.24	**3.52**	**3.42**	**34.8**	**32.8**	23.6	8.8

Our findings indicate that PG-Original gravitates towards early source sentences, while PG-MMR searches beyond the first few sentences.

In Figure 5.2 we show the median location of the first occurrences of summary n-grams, where the n-grams can come from the 1st to 5th summary sentence. For PG-Original summaries, n-grams of the 1st summary sentence frequently come from the 1st and 2nd source sentences, corresponding to the lower/higher quartiles of source sentence indices. Similarly, n-grams of the 2nd summary sentence come from the 2nd to 7th source sentences. For PG-MMR summaries, the patterns are different. The n-grams of the 1st and 2nd summary sentences come from source sentences of the range (2, 44) and (6, 53), respectively. Our findings suggest that PG-Original tends to treat the input as a single-document and identifies summary-worthy content from the beginning of the input, whereas PG-MMR can successfuly search a broader range of the input for summary content. This capability is crucial for multi-document input where important content can come from any article in the set.

Degree of extractiveness. Table 5.3 shows the percentages of summary n-grams (or entire sentences) appearing in the multi-document input. PG-Original and PG-MMR summaries both show a high degree of extractiveness, and similar findings have been revealed by See et al. [11]. Because PG-MMR relies on a handful of representative source sentences and mutes the rest, it appears to be marginally more extractive than PG-Original. Both systems encourage generating summary sentences by stitching together source sentences, as about 52% and 41% of the summary sentences do not appear in the source, but about 90% the n-grams do. The Extract+Rewrite summaries (§5.1.4), generated by rewriting selected source sentences to title-like summary sentences, exhibits a high degree of abstraction, close to that of human abstracts.

Table 5.5: Example system summaries and human-written abstract.

Human Abstract

- Boeing 737-400 plane with 102 people on board crashed into a mountain in the West Sulawesi province of Indonesia, on Monday, January 01, 2007, killing at least 90 passengers, with 12 possible survivors.
- The plane was Adam Air flight KI-574, departing at 12:59 pm from Surabaya on Java bound for Manado in northeast Sulawesi.
- The plane crashed in a mountainous region in Polewali, west Sulawesi province.
- There were three Americans on board, it is not know if they survived.
- The cause of the crash is not known at this time but it is possible bad weather was a factor.

Extract+Rewrite Summary

- Plane with 102 people on board crashes.
- Three Americans among 102 on board plane in Indonesia.
- Rescue team arrives in Indonesia after plane crash.
- Plane with 102 crashes in West Sulawesi, killing at least 90.
- No word on the fate of Boeing 737-400.
- Plane carrying 96 passengers loses contact with Makassar.
- Plane crashes in Indonesia , killing at least 90.
- Indonesian navy sends two planes to carry bodies of five.
- Indonesian plane carrying 102 missing.
- Indonesian lawmaker criticises slow deployment of plane.
- Hundreds of kilometers plane crash.

PG-Original Summary

- Adam Air Boeing 737-400 crashed Monday after vanishing off air traffic control radar screens between the Indonesian islands of Java and Sulawesi.
- Up to 12 people were thought to have survived, with rescue teams racing to the crash site near Polewali in West Sulawesi , some 180 kilometres north of the South Sulawesi provincial capital Makassar.
- It was the worst air disaster since Sept. 5, 2005, when a Mandala Airline's Boeing 737-200 crashed shortly after taking off from the North Sumatra's airport, killing 103 people.
- Earlier on Friday, a ferry carrying 628 people sank off the Java coast.

PG-MMR Summary

- The Adam Air Boeing 737-400 crashed Monday afternoon, but search and rescue teams only discovered the wreckage early Tuesday.
- The Indonesian rescue team arrived at the mountainous area in West Sulawesi province where a passenger plane with 102 people onboard crashed into a mountain in Polewali, West Sulawesi province.
- Air force rear commander Eddy Suyanto told-Shinta radio station that the plane – operated by local carrier Adam Air – had crashed in a mountainous region in Polewali province on Monday.
- There was no word on the fate of the remaining 12 people on board the boeing 737-400.

The sentences are manually de-tokenized for readability.

Linguistic quality. To assess the linguistic quality of various system summaries, we employ Amazon Mechanical Turk human evaluators to judge the summary quality, including PG-MMR, LexRank, PG-Original, and Extract+Rewrite. A turker is asked to rate each system summary on a scale of 1 (worst) to 5 (best) based on three evaluation criteria: ***informativeness*** (to what extent is the meaning expressed in the ground-truth text preserved in the summary?), ***fluency*** (is the summary grammatical and well-formed?), and ***non-redundancy*** (does the summary successfully avoid repeating information?). Human summaries are used as the ground-truth. The turkers are also asked to provide an overall ranking for the four system summaries. Results are presented in Table 5.4. We observe that the LexRank summaries are highest-rated on fluency. This is because LexRank is an extractive approach, where summary sentences are directly taken from the input. PG-MMR is rated as the best on both informativeness and non-redundancy. Regarding overall system rankings, PG-MMR summaries are frequently ranked as the 1st- and 2nd-best summaries, outperforming the others.

Example summaries. In Table 5.5 we present example summaries generated by various systems. PG-Original cannot effectively identify important content from the multi-document input. Extract+Rewrite tends to generate short, title-like sentences that are less informative and carry substantial redundancy. This is because the system is trained on the Gigaword dataset [7] where the target summary length is 7 words. PG-MMR generates summaries that effectively condense the important source content.

5.1.6 Conclusion

We describe a novel adaptation method to generate abstractive summaries from multi-document inputs. Our method combines an extractive summarization algorithm (MMR) for sentence extraction and a recent abstractive model (PG) for fusing source sentences. The PG-MMR system demonstrates competitive results, outperforming strong extractive and abstractive baselines.

5.2 Modeling Endorsement for Multi-Document Abstractive Summarization

A critical difference between single and multi-document summarization is how saliency is defined. For the latter, essential information is repeated across multiple documents, which creates an endorsement effect that increases salience of the information. However, neither the endorsement effect nor frequency is adequately modeled by modern deep neural methods. In this section, we attempt to model cross-document endorsements and their frequency for abstractive summarization, where a synopsis is created from a source document to serve as an endorser to identify salient information from other documents. Our method determines segment saliency based on its containing document and cross-document endorsements; such salience is then used to enrich an encoder-decoder model to consolidate strongly endorsed text segments into an abstract. Crucially, our method learns from fewer examples comparing to previous work, which is a very desirable characteristic. It alleviates the need for costly retraining when the set of source documents are dynamically adjusted. We validate the method on both summarization benchmarks and a new multi-document summarization dataset. In our case study, we discuss challenges and shed light on this promising direction of research. [6]

5.2.1 Introduction

"Repeat a lie often enough and it becomes the truth." Such a statement stresses the importance of *frequency* and *repetition* in comprehension, as they have an endorsement effect that increases the salience of repeated information. In this section, we make use of the endorsement effect to summarize multiple documents (MDS) that discuss a particular topic or event. In the commercial arena, its use cases include aggregating search results or distilling insights from customer reviews and user-generated content [168]. MDS is further an integral part of the work of intelligence ana-

[6] This section is adapted from: L. Lebanoff and F. Liu, Modeling Endorsement for Multi-Document Abstractive Summarization, in submission.

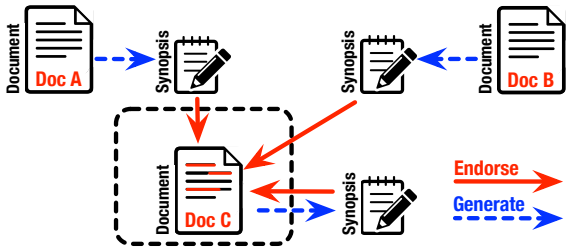

Figure 5.3: An example of synopsis-document relationships. Synopsis-document endorsements are leveraged to identify important text segments from a source document (e.g., Doc C). Strongly endorsed segments of all documents are consolidated into an abstractive summary.

lysts, who sift through a set of raw documents to identify important information, and consolidate the information into a summary to be disseminated to leadership [169].

To date, less effort has been devoted to multi-document abstractive summarization (**MuDAS**) than its single-document counterpart, despite the promising progress achieved in recent years [11, 46, 22, 47, 14]. Not only is MuDAS an important and intellectually challenging human task, but it poses a substantial challenge to modern deep neural methods—when the source documents pertaining to a topic are concatenated into a flat sequence, as is the case with many neural sequence-to-sequence models, it greatly exceeds the maximum sequence length permitted by any GPU/TPU memory.

Several recent works tackle this problem and propose techniques to encode the source documents in a hierarchical manner [50], use memory-compressed attention [170], or select representative sentences from the source documents to reduce the length of the source input [133, 109, 122].

MuDAS remains an unsolved task despite these impressive efforts. When a representative sentence "*World leaders join to pledge $8 billion for vaccine, but the U.S. sits out*" is selected from the source documents, it remains unclear which of its fragments, "*$8 billion*" or "*U.S. sits out*," is more salient given the topic of discussion and thus should be included in the abstract. Existing neural text summarization methods can fail to link frequency to text salience, partly because repetition is rarely observed in single documents, from which the models are pretrained. Further, frequency may be misrepresented as these models treat "*$8 billion*" and other quantities such as "*$5 million*" indiscriminately [25]. What has been missing so far in MuDAS is thus segment-level,

fine-grained salience modeling to achieve both fluency and information accuracy. Without that, a neural abstractive summarizer may continue to favor fluency over information accuracy and miss out on salient details.

In this work, we present a conceptual framework that leverages the endorsement effect to model fine-grained segment salience for MuDAS. A segment's endorsement score comes from the synopses of the source documents. When an analyst reads a document, he retains a synopsis of the key ideas of the document in his mind. Text segments in other documents that reiterate that synopsis are deemed salient because repetition leads to endorsement and retention [171]. We call the synopsis of the document an "**Endorser**" and the document a "**Candidate**." Segments of a candidate document that are frequently endorsed by synopses suggest high salience and they are to be consolidated into an abstract of the source documents. Our synopses are obtained from a state-of-the-art neural abstractive summarizer [14] and a variety of methods are investigated to model fine-grained endorsement between a synopsis and a document. Figure 5.3 illustrates this synopsis-document endorsement.

Our research contributions are summarized as follows. (*i*) We introduce a conceptual framework to model asynchronous endorsement between synopses and documents for MuDAS. (*ii*) We present a new summarization method to enrich a neural encoder-decoder model with fine-grained endorsement to enable the model to consolidate strongly endorsed segments into an abstract. (*iii*) We demonstrate the effectiveness of our method by conducting extensive experiments on benchmark MuDAS datasets and a large, newly-introduced multi-document summarization dataset and performing a case study. All of our models and source code will be released publicly to shed light on this complex task.

5.2.2 Summarizing with Endorsement

We are particularly interested in condensing multiple source documents into a single document, then consolidate the content into an abstract [19, 109]. We enhance the single document with

fine-grained segment salience to offset the lead bias [18, 110], which hinders the development of multiple-document summarization. Our salience estimates are obtained from a frequency-driven endorsement model. Frequency and redundancy are essential in multi-document summarization. Without these, even humans tend to disagree on what information is relevant and should be retained in the summary [111]. In what follows, we present details of our summarization framework.

We approach the MuDAS problem in two stages. First, we obtain fine-grained segment-level endorsement for any candidate document. It allows for a significant reduction of source texts from multiple documents into a single *mega-document*—a pseudo-document with any unendorsed text eliminated from consideration. We next present a state-of-the-art abstractive summarizer to reduce the mega-document to an abstract, analogously to how an editor would consolidate texts with an emphasis on endorsed content. The process involves non-trivial design decisions. In the following, we start by presenting our summarization architecture with endorsement.

We favor the encoder-decoder architecture over its decoder-only counterpart [48, 154, 172]. The architecture allows us to weigh the impact of source texts and the segment-level endorsement in summary generation. The encoder and decoder each comprise of a stack of L Transformer blocks [134]. Let $\{x\}_{i=0}^{m}$ be the source sequence consisting of the mega-document, and $\{y\}_{j=0}^{n}$ the summary sequence. Let $\boldsymbol{E}$ be a matrix of token embeddings and $\boldsymbol{P}$ be token position embeddings. An encoder produces a set of hidden vectors in its l-th layer (Eq. (5.11)), $\boldsymbol{H}^{(l)} = \langle \boldsymbol{h}_0^{(l)}, \ldots, \boldsymbol{h}_m^{(l)} \rangle$, where $\boldsymbol{h}_i^{(l)}$ is a hidden vector of the i-th source token. A decoder utilizes hidden vectors of the top layer, $\boldsymbol{H}^{(L)}$, to build representations for the summary sequence, where $\boldsymbol{G}^{(l)}$ represents a sequence of hidden vectors of the l-th layer (Eq. (5.12)). A upper triangular-shaped mask is used by the

decoder, so that $\boldsymbol{g}_j^{(l)}$ only depends on summary tokens whose positions are less than j.

$$\boldsymbol{H}^{(l)} = \langle \boldsymbol{h}_0^{(l)}, \ldots, \boldsymbol{h}_m^{(l)} \rangle \tag{5.11}$$
$$= \begin{cases} \langle \boldsymbol{E}_{x_0} + \boldsymbol{P}_0, \ldots, \boldsymbol{E}_{x_m} + \boldsymbol{P}_m \rangle & l = 0 \\ \textsc{EncBlock}_l\big(\boldsymbol{H}^{(l-1)}\big) & l > 0 \end{cases}$$

$$\boldsymbol{G}^{(l)} = \langle \boldsymbol{g}_0^{(l)}, \ldots, \boldsymbol{g}_n^{(l)} \rangle \tag{5.12}$$
$$= \begin{cases} \langle \boldsymbol{E}_{y_0} + \boldsymbol{P}_0, \ldots, \boldsymbol{E}_{y_n} + \boldsymbol{P}_n \rangle & l = 0 \\ \textsc{DecBlock}_l\big(\boldsymbol{G}^{(l-1)}, \boldsymbol{H}^{(L)}\big) & l > 0 \end{cases}$$

We argue that, with this architecture, it is preferable to modify the decoder to bias it toward endorsed content during decoding, rather than modifying the encoder, as encoder representations $\boldsymbol{H}^{(L)}$ can often be unsupervisedly pretrained. It would be best if such representations remain unaffected by whether or not a piece of text is endorsed to provide maximum model flexibility. A decoder layer consists of three main blocks to transform from $\boldsymbol{G}^{(l-1)}$ to $\boldsymbol{G}^{(l)}$ (Eqs. (5.13-5.15)).[7] In particular, self-attention allows a summary token to attend to other tokens prior to it. Cross-attention allows a summary token to attend to all source tokens using $\boldsymbol{H}^{(L)}$. Finally, a feed-forward network consisting of two linear transformations with a ReLU in between is applied to generate $\mathbf{G}^{(l)}$. Our focus of this work is thus to adjust the cross-attention to subtly emphasize on endorsed content during decoding.

$$\widetilde{\boldsymbol{G}}^{(l-1)} = \textsc{Self-Attn}(\boldsymbol{G}^{(l-1)}) \tag{5.13}$$
$$\widetilde{\boldsymbol{G}}^{(l)} = \textsc{Cross-Attn}(\widetilde{\boldsymbol{G}}^{(l-1)}, \boldsymbol{H}^{(L)}) \tag{5.14}$$
$$\mathbf{G}^{(l)} = \textsc{FeedForward}(\widetilde{\boldsymbol{G}}^{(l)}) \tag{5.15}$$

An *original* cross-attention head z transforms the decoder $\widetilde{\boldsymbol{g}}_j^{(l-1)}$ and encoder $\boldsymbol{h}_i^{(L)}$ respectively into query, key and value vectors (Eqs. (5.16-5.18)), then computes attention weights as normalized

[7] We omit the residual connection and layer normalization associated with each block for brevity.

dot products between the query and key vectors. The output of the head is a weighted sum of the value vectors.

$$\boldsymbol{q}_j^z = \boldsymbol{W}_z^Q \widetilde{\boldsymbol{g}}_j^{(l-1)} \qquad j \in [n] \tag{5.16}$$

$$\boldsymbol{k}_i^z = \boldsymbol{W}_z^K \boldsymbol{h}_i^{(L)} \qquad i \in [m] \tag{5.17}$$

$$\boldsymbol{v}_i^z = \boldsymbol{W}_z^V \boldsymbol{h}_i^{(L)} \qquad i \in [m] \tag{5.18}$$

$$M_i^\tau = \begin{cases} 1 & \text{if Endorse}(x_i) \geq \tau \\ 0 & \text{otherwise} \end{cases} \tag{5.19}$$

$$\text{head}_j^{z,\tau} = \sum_{i=0}^{m} \frac{\exp(\boldsymbol{q}_j^{z\top} \boldsymbol{k}_i^z)}{\sum_{r=0}^{m} \exp(\boldsymbol{q}_j^{z\top} \boldsymbol{k}_r^z)} M_i^\tau \boldsymbol{v}_i^z \tag{5.20}$$

$$\widetilde{\boldsymbol{g}}_j^{(l)} = \sum_{z=1}^{n_{\text{head}}} \sum_{\tau=0}^{\tau_{\max}} \text{head}_j^{z,\tau} \boldsymbol{W}_z^\tau \tag{5.21}$$

Importantly, we introduce a set of *companion heads* for each original head, where all companion heads of z share its parameters $\{\boldsymbol{W}_z^Q, \boldsymbol{W}_z^K, \boldsymbol{W}_z^V\}$ but $\text{head}_j^{z,\tau}$ attends only to source tokens that are endorsed τ times or more, achieved through masking (Eqs. (5.19-5.20)). An original head is believed to copy content from source tokens, which are deemed relevant to the j-th summary token according to dependency syntax or coreference [1]. A companion head serves a similar purpose but has a narrower focus on endorsed tokens—frequently endorsed source tokens will be captured by companion heads of varying τ values. Finally, all heads are pooled into a hidden vector $\widetilde{\boldsymbol{g}}_j^{(l)}$ (Eq. (5.21)) to be passed to the feedforward layer, where frequently endorsed content can easily manifest itself.

When $\tau_{\max}$ is set to 0, the model reduces to its initial form and only the original heads are retained, i.e., $\text{head}_j^{z,0}$. Further, we initialize $\boldsymbol{W}_z^\tau = \lambda^\tau \boldsymbol{W}_z$, where $\boldsymbol{W}_z \in \mathbb{R}^{h_{\text{head}} \times h_{\text{model}}}$ are pretrained model parameters associated with the head z; $\lambda^\tau \in [0,1]$ is a coefficient and $\boldsymbol{W}_z = \sum_{\tau=0}^{\tau_{\max}} \boldsymbol{W}_z^\tau$. It indicates that, head z and all of its companion heads are linearly interpolated to produce the decoder hidden vector $\widetilde{\boldsymbol{g}}_j^{(l)}$. If any source token is unendorsed, it will have a reduced impact on the decoder hidden vector when companion heads are used. When frequency of endorsement is dy-

namically adjusted, our model can avoid costly retraining by adjusting the level of endorsement (τ). We proceed by describing how fine-grained token-level endorsement is obtained from modeling synopsis-document relationships.

5.2.3 Modelling Endorsement for MuDAS

Modelling endorsement serves two main purposes. It identifies salient segments of text that are used to direct the summarizer toward consolidating salient information. Further, it allows us to significantly reduce the amount of source texts required by the summarizer from multiple documents to a single mega-document, where any unendorsed texts can be eliminated from consideration.

A fragment of text is considered to be endorsed if its information is observed in the endorser. We obtain a set of synopses from the source documents; they are used as *endorsers* to identify salient segments from a candidate source document. See Table 5.11 for an example candidate document with highlighted text segments indicating endorsement. A segment that is endorsed only once indicates its information is deemed important by only one source document. Frequent endorsement by multiple endorsers suggests the information is reiterated in multiple source documents, and reiteration implies increased salience. Any information that is present among multiple sources is likely to be important. Thus, our method identifies salient segments considering both within- and cross-document saliency. Our approach is in spirit similar to those of building semantic concept graphs for multi-document summarization [73, 173, 21] in that frequently reiterated concepts are likely to be captured. However, we do not explicitly construct semantic concept graphs, but focus on modeling synopsis-document endorsement and incorporating it into summary generation, which distinguishes our work from these studies. We investigate two variants to compute segment-level endorsement.

5.2.3.1 Synopsis-Document Alignment

Let S be a synopsis serving as the endorser and D a source document, our goal is to estimate whether a token x_i of the document is endorsed by the synopsis. A soft alignment between the synopsis and document is attainable by utilizing text evaluation metrics such as BERTScore [92], where we build contextualized embeddings for tokens of the document and synopsis, compute the cosine similarity of embeddings, and find a most similar synopsis token for each token of the document to obtain the endorsement score $S(x_i)$ (Eq. (5.22)). Albeit a greedy alignment, the method can produce competitive results comparing to methods such as the earth mover's distance [174].

$$S(x_i) = \max_{y_j \in S} \text{Sim}(x_i, y_j) \tag{5.22}$$

$$\{s, e\} = \arg\max_{\{i,j\} \in m} \sum_{k=i}^{j} (S(x_k) - \delta) \tag{5.23}$$

Contiguous Segments It is important to endorse segments of text rather than isolated tokens, as segments such as "*$8 million*" is either included in the abstract in its entirety, or not at all. We transform token-level endorsement scores into binary decisions using the maximum sum subarray algorithm (Eq. (5.23)), which finds a contiguous subsequence that yields the highest sum of scores. The solution is trivial when all scores are positive. We thus offset the scores by δ before applying the algorithm. Let $\{0.2, 0.3, -0.1, 0.4, -0.5\}$ be an example of a set of adjusted endorsement scores, the algorithm endorses the first four tokens as the sum of their scores is the highest, yielding $\{1, 1, 1, 1, 0\}$, where 1 indicates the token is endorsed and 0 otherwise. We apply the algorithm to each sentence of the document and discard the segment if it has less than 5 tokens. The method endorses salient segments of text, yet is lenient to include gap tokens.

Soft vs. Hard Alignment A hard alignment between the synopsis and document can be obtained from string matching. A document token receives a score of 1 if it finds a match in the synopsis. Similar to above, we offset the scores by δ to obtain segments of endorsed text. Hard alignment is

sensitive to entities and quantities; yet it can miss out on paraphrases. We compare the effectiveness of these alignment methods in the results section.

5.2.3.2 Synopses as Endorsers

A synopsis contains the main points of the source document. We employ BART [14] as a single-document abstractive summarizer to produce a synopsis from each document of the input cluster. Synopses as endorsers are superior to whole documents or sentence extracts. Not only are synopses more concise, but they can exclude superfluous information such as quoted material from consideration. We score all sentences of the source documents according to the sum of their token endorsement scores. Highest endorsed sentences are selected and arranged in chronological order to form a mega-document, with a limit of $|D|$ tokens, which serves as the input to our MuDAS summarization module.

When a token is deemed salient by τ endorsers, we set Endorse(x_i)=τ, analogous to a majority vote by the pool of endorsers. We introduce *reciprocal endorsement*, where a synopsis can endorse every document of the cluster; and *sequential endorsement*, where source documents are arranged in chronological order and only synopses of the later documents can endorse the earlier documents. Sequential endorsement assumes the first few articles of an event or topic are more important than others; it avoids endorsing redundant content, which is particularly useful when the documents contain redundancy or noise that is typical in the output of clustering algorithms for content aggregation.

Table 5.6: Statistics of our datasets.

Dataset	SynopLen	#Segments	SegLen
WCEP	61	4.9	14.2
DUC-04	58	6.1	11.7
TAC-11	60	6.7	11.8

We show the average length of synopses (**SynopLen**), the average number of segments in a source document endorsed by a single synopsis and the average length of endorsed segments (**SegLen**).

5.2.4 Data

We experiment with a large-scale multi-document summarization dataset [123] whose data are gathered from the Wikipedia Current Events Portal (WCEP).[8] The dataset contains an archive of important news events happening around 2016–2019. Each event is associated with a concise summary of 30-40 words written by an editor and an average of 1.2 source articles linked from the event page. Additional source articles are retrieved from the CommonCrawl-News dataset using an event classifier. These articles are likely related to the event and published within a window of ± 1 day of the event date. We sample from these additional articles to ensure each event has 10 source articles. All summaries and source articles are in English. The dataset contains 8,158, 1,020 and 1,022 clusters respectively in the train, validation and test splits.

Our method aims to produce an abstractive summary from a cluster of news articles discussing a particular event or topic. To assess the generality of our method, we apply the model (trained on WCEP) to three test sets, including the test split of WCEP and two benchmark multi-document summarization datasets, DUC-04 and TAC-11. The DUC/TAC datasets contain 50 and 44 clusters, respectively. They each comprise a set of news events collected over different periods of time, and thus are suitable for evaluation of the model's generality in out-of-domain scenarios. DUC/TAC datasets contain four reference summaries per cluster created by NIST assessors. WCEP has a single reference summary per cluster written by human editors. The maximum summary length

[8] `https://en.wikipedia.org/wiki/Portal:Current_events`

is 100 words for DUC/TAC and 40 words for WCEP, following previously published conventions. Additional statistics for each dataset are presented in Table 5.6.

5.2.5 Experimental Setup

Baseline Systems. We compare our endorsement method to several strong baselines on multi-document summarization. Our extractive baseline systems include (i) *TextRank* [175] and *LexRank* [139], which are graph-based summarization approaches estimating sentence importance based on eigenvector centrality; (ii) *Centroid* [163] computes the importance of each source sentence based on its cosine similarity with the document centroid; (iii) *Submodular* [176] treats multi-document summarization as a submodular maximization problem and uses a submodular function to estimate summary importance; (iv) *KL-Sum* [102] is a greedy approach that adds sentences to the summary to minimize KL divergence. (v) *TSR* and *BertReg* [123] are regression-based sentence ranking methods using statistical features and averaged word embeddings (TSR) and with sentence embeddings computed by a pretrained BERT model (BertReg). Moreover, our neural abstractive baseline systems include: (vi) *PointerGen* [11] learns to generate by reusing source words or predicting new words; the documents are concatenated to form the input sequence. (vii) *PG-MMR* [133] exploits the maximal marginal relevance method to select sentences and an encoder-decoder model fuses the sentences into an abstract; (viii) *Hi-MAP* [122] introduces an end-to-end hierarchical attention model to generate abstracts from multi-document inputs. We compare our system to these baselines and report results on WCEP, DUC-04, and TAC-11 datasets[9].

Sequential vs. Reciprocal Endorsement. We investigate two variants of our model: (a) *reciprocal endorsement* allows any two documents of the same cluster to endorse each other, and (b) *sequential endorsement* arranges source documents in chronological order and only later documents are allowed to endorse earlier ones to avoid redundancy. For both variants, the highest-

[9]We were unable to compare our method with hierarchical Transformers [50] because the authors did not make their ranker available for ranking paragraphs. Additionally, their model was trained on the WikiSum dataset rather than news corpora, hence there is a domain mismatch problem.

Table 5.7: Percentage of tokens above endorsement score threshold.

	% endorse scores $\geq \tau$		
Dataset	$\tau = 0$	$\tau = 1$	$\tau = 2$
WCEP	100.0	12.6	5.6
DUC-04	100.0	9.7	2.3
TAC-11	100.0	14.5	4.1

Percentage of tokens with endorsement scores above each threshold value used in each set of companion heads. All tokens with scores below the threshold are masked out.

scoring sentences are consolidated to form a mega-document which, along with the endorsement scores, are passed to our endorsement-aware abstractor to be condensed into a summary.

Endorsement-Aware Abstractor. We employ BART, a state-of-the-art encoder-decoder model as our base abstractor [14]. The model has 12 layers of Transformers and 16 attention heads in each of the encoder and decoder. It uses a hidden size of 1024. The model was fine-tuned on the train split of WCEP for an average of two epochs with a batch size of 4. We use the Adam optimizer [177] and a learning rate of 3^{-5} with warm-up. At inference time, we use a beam size of K=4, with a minimum decoding length of 10 and a maximum of 50 tokens. Our implementation is based on fairseq[10] and it takes about two hours to train the model on a NVIDIA V100 32GB GPU card.

For the endorsement-aware abstractor, we add two sets of companion heads to the decoder, for a total of 48 attention heads. The τ values for each set of heads are 0, 1, and 2. Table 5.7 shows the percentage of tokens that receive different levels of attention: 12% of the tokens receive level-1 attention ($\tau = 1$), 4% receive level-2 attention ($\tau = 2$). The λ^{τ} values are set to be 0.8, 0.1, and 0.1—this gives more influence to the original attention heads, so the model is not confused by the addition of the new heads that attend to endorsed segments. We use a maximum of 1024 tokens for the mega-document, which is used as input to the BART model.

Synopsis-Document Endorsement. To enable soft alignment between a synopsis and a candidate document, we use BERTScore [92] with the following hash code: roberta-large_L17_no-

[10] `https://github.com/pytorch/fairseq`

Table 5.8: A comparison of multi-document summarization methods on the WCEP test set.

System	R-1	R-2	R-SU4
Extractive			
Random Lead	27.6	9.1	–
Random	18.1	3.0	–
TextRank	34.1	13.1	–
Centroid	34.1	13.3	–
Submodular	34.4	13.1	–
TSR	35.3	13.7	–
BertReg	35.0	13.5	–
Our Method			
Endorser-Reciprocal	43.3	21.9	22.1
Endorser-Sequential	**45.4**	**23.2**	**23.5**

*Endorser-** are our proposed methods. Previous work on WCEP does not report R-SU4 results and are thus left out.

idf_version=0.3.2(hug_trans=2.8.0)-rescaled. It suggests that the token representations are drawn from the 17th layer of RoBERTa-large. Our maximum sum subarray algorithm (Eq. (5.23)) requires the scores to contain a mix of positive/negative values. Thus, we subtract all scores by δ. The δ values are 0.85 and 0.8 for the soft alignment and hard alignment, respectively. These values are tuned on validation data, where a larger δ indicates fewer tokens will be endorsed, and vice versa.

We proceed by presenting summarization results on our datasets, including an ablation study to examine the contribution of each part of our method. We then present a case study showcasing the potential of our endorsement method.

5.2.6 Results

Our methods achieve state-of-the-art when compared to previously reported baselines on WCEP (Table 5.8). Sequential endorsement outperforms reciprocal endorsement due to the ability of sequential endorsement to remove redundancies introduced in later documents. In news domain,

Table 5.9: A comparison of multi-document summarization methods on the DUC-04 dataset.

System	R-1	R-2	R-SU4
Extractive			
TextRank	33.16	6.13	10.16
LexRank	34.44	7.11	11.19
Centroid	35.49	7.80	12.02
Neural Abstractive			
Pointer-Gen	31.43	6.03	10.01
PG-MMR	36.88	8.73	12.64
PG-BRNN	29.47	6.77	7.56
Hi-MAP	35.78	8.90	11.43
Our Method			
Endorser-Sequential	34.74	8.08	12.06
Endorser-Reciprocal	35.24	8.61	12.49
Endorser-Oracle	36.27	8.93	13.04

*Endorser-** are our methods.

later articles generally review information from previous articles and introduce small developments in the story. By ordering the documents chronologically and having later articles give endorsement to earlier articles, it encourages the summarizer to pick content from earlier articles and reduce redundancy introduced in later articles. The largest performance increase can be seen in R-2, with *Endorser-Sequential* achieving a 9.7 increase over a BERT-based method. It demonstrates the effectiveness of endorsement for detecting salient segments and stitching them together to form a summary.

We present results on DUC-04 and TAC-11 datasets in Tables 5.9 and 5.10. Here, our methods either outperform or perform comparably to previous summarization methods. Note that our model is trained on the train split of WCEP and it is tested under different scenarios. On the WCEP test set, it corresponds to an in-domain scenario. On the DUC-04 and TAC-11 datasets, it is an *out-of-domain* scenario, as these datasets are collected over different periods of time. The fact that our system, when used out-of-the-box, can attain better or comparable results to the previous state-of-the-art on DUC-04 and TAC-11 datasets has demonstrated its strong generalization capability. It

Table 5.10: A comparison of multi-document summarization methods on the TAC-11 dataset.

System	R-1	R-2	R-SU4
Extractive			
KLSumm	31.23	7.07	10.56
LexRank	33.10	7.50	11.13
Neural Abstractive			
Pointer-Gen	31.44	6.40	10.20
PG-MMR	37.17	10.92	14.04
Our Method			
Endorser-Sequential	36.11	9.52	13.07
Endorser-Reciprocal	37.43	10.71	13.94
Endorser-Oracle	38.01	11.11	14.61

*Endorser-** are our methods.

suggests that obtaining segment-level endorsement on an outside domain then using it to inform summary generation is meaningful.

Intuitively, we want to steer the model attention towards endorsed segments if they are of high quality, and away from the segments otherwise. We conduct a set of oracle experiments that set λ^{τ} values to be proportional to the R-2 recall scores of endorsed segments. If the segments obtained for $\tau = 2$ yield a high R-2 recall score, they contain summary content and the model should attend to these endorsed segments by using a high λ^{τ} value. Results are reported in Tables 5.9 and 5.10 (*Endorser-Oracle*). We find that such a strategy is effective for making the most of companion heads. Future work may associate attention (λ^{τ} values) with the quality of segments obtained at different levels of endorsement ($\tau = \{0, 1, 2\}$).

We observe that the reciprocal endorsement strategy outperforms the sequential endorsement for DUC-04 and TAC-11. A closer look at the data suggests that this is partly due to the lower amount of redundancy present in DUC/TAC data. While WCEP documents are automatically clustered and contain much redundancy, the source documents of DUC/TAC are manually selected by NIST assessors, each successive document in a topic cluster presents new developments about the topic. Thus, reciprocal endorsement may lead to better results for domains with less redundancy.

Table 5.11: Endorsed segments for a document.

(a) Single Synopsis Generated by BART

Opposition leader Sam Rainsy seeks clarification of security guarantees promised by Hun Sen. Hun Sen announced a government guarantee of all politicians' safety Wednesday. The opposition leader was forced to take refuge in a U.N. office in September to avoid arrest. The two parties have formed three working groups to hammer out details of the agreement.

(b) Endorsement from All Synopses

Sam Rainsy, who earlier called Hun Sen's statement "full of loopholes," asked Sihanouk for his help in obtaining a promise from Hun Sen that all members of the Sam Rainsy Party were free from prosecution for their political activities during and after last July's election. Sam Rainsy, a staunch critic of Hun Sen, was forced to take refuge in a U.N. office in September to avoid arrest after Hun Sen accused him of being behind a plot against his life. The alleged assassination attempt came during massive street demonstrations organized by the opposition after Hun Sen's Cambodian People's Party narrowly won the election. The opposition, alleging widespread fraud and intimidation, refused to accept the results of the polls. Fearing for their safety, Sam Rainsy and his then-ally Prince Norodom Ranariddh led an exodus of opposition lawmakers out of Cambodia after parliament was ceremonially opened in late September. Ranariddh, whose FUNCINPEC party finished a close second in the election, returned last week and struck a deal with Hun Sen to form a coalition government. The agreement will make Hun Sen prime minister and Ranariddh president of the National Assembly. The two parties have formed three working groups to hammer out details of the agreement, including the establishment of a Senate to be the upper house of parliament. Sok An, representing Hun Sen's party, said...

(c) Human-Chosen Segments

Sam Rainsy, who earlier called Hun Sen's statement "full of loopholes," asked Sihanouk for his help in obtaining a promise from Hun Sen that all members of the Sam Rainsy Party were free from prosecution for their political activities during and after last July's election. Sam Rainsy, a staunch critic of Hun Sen, was forced to take refuge in a U.N. office in September to avoid arrest after Hun Sen accused him of being behind a plot against his life. The alleged assassination attempt came during massive street demonstrations organized by the opposition after Hun Sen's Cambodian People's Party narrowly won the election. The opposition, alleging widespread fraud and intimidation, refused to accept the results of the polls. Fearing for their safety, Sam Rainsy and his then-ally Prince Norodom Ranariddh led an exodus of opposition lawmakers out of Cambodia after parliament was ceremonially opened in late September. Ranariddh, whose FUNCINPEC party finished a close second in the election, returned last week and struck a deal with Hun Sen to form a coalition government. The agreement will make Hun Sen prime minister and Ranariddh president of the National Assembly. The two parties have formed three working groups to hammer out details of the agreement, including the establishment of a Senate to be the upper house of parliament. Sok An, repr Hun Sen's party, said...

A synopsis **(a)** is generated from the candidate document, which is used to then endorse segments from the document. The document also receives endorsement from the other 9 synopses in the cluster **(b)**. We compare to segments chosen by a human using the Pyramid method **(c)**. Stronger highlighting indicates the segment received endorsement from many synopses.

5.2.6.1 Ablation

We perform an ablation study on WCEP to study the effects of each component in our model (Table 5.12). First, we compare the endorsement methods, denoted by *HardAlign* and *SoftAlign*. SoftAlign achieves consistently better results, showing that it is important to allow some flexibility when aligning synopses to documents for endorsement. Next, we remove several components from the best-performing model (SoftAlign) to understand the effect of each. Removing "companion heads" from the abstractive model results in a very small boost in performance. Removing "endorsement selection"—meaning the model uses no information gained from performing endorsement, and is simply a BART model trained to summarize documents—leads to a significant performance drop, especially in R-1. It suggests that using endorsement to identify summary-worthy content from multiple documents is beneficial for an abstractive model.

Moreover, removing the "abstractive model"—meaning summaries are created extractively by selecting the highest-endorsed sentences—results in a large decrease in scores. It indicates that content-selection by endorsement cannot be done alone without an abstractor to create a more concise summary. This is especially the case for the WCEP dataset, where human reference summaries are relatively short.

We additionally report BERTScore [92] to evaluate summaries, in addition to the ROUGE metric [97]. BERTScore uses cosine similarity between BERT contextual embeddings of words to detect word overlap between two texts, thus overcoming the problem of lexical variation in summarization. On DUC-04, the F_1 scores are 29.89 and 30.14, respectively for our sequential and reciprocal model. The score for the human reference summary is 35.08. This shows very similar trends to those in Table 5.9. Our method when tested in out-of-domain scenarios can achieve competitive scores in comparison to strong baselines.

Table 5.12: Ablation study on WCEP dataset.

System	R-1	R-2	R-SU4
Endorser-HardAlign	44.7	22.4	22.6
Endorser-SoftAlign	45.4	23.2	23.5
- companion heads	45.8	23.5	23.8
- endorse selection	43.6	23.0	22.9
- abstractive module	28.3	9.3	10.9

5.2.7 A Case Study

We present a human evaluation of our fine-grained endorsement in Table 5.11. Text segments are endorsed using a soft alignment between the candidate document and synopses. We compare the resulting endorsements to the text segments chosen by a human, using the Pyramid method [178], where Summarization Content Units (SCUs) are identified from the reference summaries and are matched to phrases in the candidate document. The segments selected by our endorsement method and those chosen by human annotation show a great amount of overlap, exemplifying the strength of our method in locating salient content from multi-document inputs. In fact, our endorsement method draws strong parallels with the Pyramid method—in our case, sentences from the automatically-generated synopses act as SCUs, which are matched to phrases in the candidate document using a soft or hard alignment.

We observe that the endorsement given by a single synopsis is already quite similar to the human segments. However, taking the average endorsement from all ten synopses results in a higher quality set of segments. This shows the inherent value that exists from repetition in multi-document clusters, and it shows the importance of leveraging all of the documents rather than just a single one for salience estimation. Importantly, we observe that named entities (e.g., *"Sam Rainsy," "King Norodom Sihanouk,"* etc.) are more readily endorsed than other phrases. The reason is because these entities are frequently repeated verbatim in all of the documents, thereby increasing their likelihood of being endorsed.

We envision a future neural document summarizer to produce better synopses than BART. It can lead to more accurate estimates for endorsed segments, hence improving the overall performance of our multi-document summarizer. The endorsement procedure at its core is simple and robust—looking for shared content between the document and each synopsis. It also provides great flexibility allowing the summarizer to potentially operate on document clusters containing a varying number of source documents, as opposed to a fixed number of documents, which is a very desirable characteristic.

5.2.8 Conclusion

In this section, we introduce a conceptual framework to model asynchronous endorsement between synopses and documents for multi-document abstractive summarization. A synopsis is created from a document to serve as an endorser to identify salient information from other documents. We present a novel summarization method to enrich an encoder-decoder model with fine-grained endorsement to enable the model to consolidate strongly endorsed segments into an abstractive summary. We validate the proposed method on both summarization benchmarks and a new multi-document summarization dataset. Finally, we discuss challenges using a case study and shed light on this promising direction of research.

CHAPTER 6: CONCLUSION

In this dissertation, we have presented several contributions to the field of text summarization. We demonstrated the efficacy of separating content selection from surface realization in summarization models. In this final chapter, we reiterate our contributions and propose future directions of research in this area.

6.1 Contributions

Throughout this dissertation, we presented multiple content selection and sentence fusion methods for both single-document and multi-document summarization.

Chapter 3 presents two content selection approaches. First, we show that scoring sentence singletons and pairs in a unified space leads to better selection of sentences. We also show that humans tend to choose one or two sentences (rarely more) to then compress or fuse into summary sentences. Second, we show that a cascaded architecture can produce further gains in content selection by selecting sentences and words/phrases in those sentences, then finally using a surface realization model to generate a summary sentence. We find that it is helpful to select a large percentage of words from the sentences, which shows it focuses on removing unimportant words rather than attending to salient words.

Chapter 4 presents a study of sentence fusion and surface realization. We analyze state-of-the-art system summaries and find a large percentage of output sentences are ungrammatical or unfaithful. This motivates further work on performing sentence fusion effectively and faithfully.

To this end, we introduce a dataset based the CNN/Daily Mail dataset containing sentence fusion examples with points of correspondence. We demonstrate that detection of points of correspondence is not trivially solved using current coreference resolution models, and that our data can be useful as a testbed for sentence fusion. We also present two methods incorporating these points of correspondence to enhance Transformer models for sentence fusion.

Chapter 5 presents two approaches for multi-document summarization. First, we attempt to adapt an encoder-decoder model from single-document summarization to multi-document summarization using Maximal Marginal Relevance (MMR) as a content selector. This approach outperforms other existing abstractive approaches, while also largely overcoming the lead bias present in most models. Second, we propose content selection of sentences based on document endorsement. A case study demonstrates the effectiveness of the method and shows significant parallels to the Pyramid method.

6.2 Future Directions

We have shown that using explicit steps of content selection and surface realization can give greater model flexibility and lead to greater summarization performance. As the summarization field matures, improvement in automatic metrics on standard datasets will begin to stagnate. The generated summaries, however, will still have issues, especially problems with coherency and factual consistency. It is thus important to create new methods of evaluation – automatic methods and manual methods performed by humans. Evaluation may also be done as a combination of automatic and manual to reduce both cost and variance in annotations.

Additionally, more research will be focused on summarization datasets with few or no labeled examples. Separating the steps of content selection from surface realization can alleviate this issue. Content selection methods do not require much data, and thus can be developed relatively easily for a specific dataset. A surface realization model requires more data but is usually more general and not tied heavily to specific domains. These models can then be trained on more abundant datasets

such as news or unlabeled data in a specific domain. We believe future research can develop new methods for content selection in new domains such as book chapters [158], legal documents [61, 117], and scientific articles [116].

Finally, multi-document summarization can make excellent use of explicit content selection and surface realization. Recently, unsupervised and few-shot learning approaches were introduced for summaries of multiple opinion reviews [179, 108]. Future work can augment these models by performing content selection, perhaps by clustering information that is repeated in reviews, and then fusing the selected content together using a surface realization model. Content selection is vital to multi-document summarization due to its nature of having very long inputs. In order to capture the essence of these large clusters of documents, methods using content selection in tandem with surface realization should be explored further.

APPENDIX: VITA

Logan Lebanoff earned the B.S. degree in Computer Science and M.S. degree in Computer Science from the University of Central Florida, Orlando, Florida, in 2016 and 2019, respectively. Since the Fall of 2016, he has been a Ph.D. student and Research Assistant in the Natural Language Processing group at the University of Central Florida. He is a recipient of the UCF Presidential Doctoral Fellowship from 2016 to 2020. During the summer of 2019, he worked as a research intern at Adobe Research in San Jose, CA. During the summer of 2020, he worked as an artificial intelligence engineering intern at SoarTech in Orlando, FL. He has accepted a full-time position as Artificial Intelligence Engineer II at SoarTech beginning in December 2020. Logan has ten publications at top NLP and AI conferences and workshops, including ACL, EMNLP, COLING, AACL, and AAAI. He has served as session chair for the EMNLP 2020 virtual conference and has been invited to serve as area chair for NAACL 2021. He has served on the program committee for reviewing papers for several conferences. Logan Lebanoff's research interests include natural language processing, machine learning, deep learning, text summarization, and natural language generation.

Publications

L. Lebanoff, J. Muchovej, F. Dernoncourt, D. S. Kim, L. Wang, W. Chang, and F. Liu. "Learning to Fuse Sentences with Transformers for Summarization." In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

L. Lebanoff, F. Dernoncourt, D. S. Kim, S. Kim, W. Chang, and F. Liu, "A cascade approach to neural abstractive summarization with content selection and fusion." In *Proceedings of the Asia-Pacific Chapter of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (AACL-IJCNLP)*, 2020.

L. Lebanoff, J. Muchovej, F. Dernoncourt, D. S. Kim, L. Wang, W. Chang, and F. Liu, "Understanding points of correspondence between sentences for abstractive summarization." In *Proceedings for the ACL 2020 Student Research Workshop*, 2019.

L. Lebanoff, J. Muchovej, F. Dernoncourt, D. S. Kim, S. Kim, W. Chang, and F. Liu, "Analyzing sentence fusion in abstractive summarization." In *Proceedings for the EMNLP 2019 Workshop on New Frontiers in Summarization*, 2019.

L. Lebanoff, K. Song, F. Dernoncourt, D. S. Kim, S. Kim, W. Chang, and F. Liu, "Scoring sentence singletons and pairs for abstractive summarization." In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

K. Song, L. Lebanoff, Q. Guo, X. Qiu, X. Xue, C. Li, D. Yu, and F. Liu. "Joint parsing and generation for abstractive summarization." *Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

S. Cho, L. Lebanoff, H. Foroosh, and F. Liu, "Improving the similarity measure of determinantal point processes for extractive multi-document summarization." In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

L. Lebanoff, K. Song, and F. Liu, "Adapting the neural encoder-decoder framework from single to multi-document summarization," In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

L. Lebanoff and F. Liu. "Automatic detection of vague words and sentences in privacy policies." In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

K. Liao, L. Lebanoff, and F. Liu, "Abstract meaning representation for multi-document summarization." In *Proceedings of the International Conference on Computational Linguistics (COLING)*, 2018.

LIST OF REFERENCES

[1] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, "What does BERT look at? an analysis of BERT's attention," in *Proc. of the 2019 ACL Workshop BlackboxNLP*, 2019.

[2] H. Daume III and D. Marcu, "A noisy-channel model for document compression," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002.

[3] D. Zajic, B. J. Dorr, J. Lin, and R. Schwartz, "Multi-candidate reduction: Sentence compression as a tool for document summarization tasks," *Information Processing and Management*, 2007.

[4] D. Gillick and B. Favre, "A scalable global model for summarization," in *Proceedings of the NAACL Workshop on Integer Linear Programming for Natural Langauge Processing*, 2009.

[5] D. Galanis and I. Androutsopoulos, "An extractive supervised two-stage method for sentence compression," in *Proceedings of NAACL-HLT*, 2010.

[6] T. Berg-Kirkpatrick, D. Gillick, and D. Klein, "Jointly learning to extract and compress," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2011.

[7] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for sentence summarization," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015.

[8] R. Nallapati, B. Zhou, C. dos Santos, C. Gulcehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence RNNs and beyond," in *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, 2016.

[9] J. Tan, X. Wan, and J. Xiao, "Abstractive document summarization with a graph-based attentional neural model," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017.

[10] R. Paulus, C. Xiong, and R. Socher, "A deep reinforced model for abstractive summarization," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017.

[11] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017.

[12] A. Celikyilmaz, A. Bosselut, X. He, and Y. Choi, "Deep communicating agents for abstractive summarization," in *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018. [Online]. Available: https://arxiv.org/pdf/1803.10357.pdf

[13] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *arXiv:1910.10683*, 2019. [Online]. Available: https://arxiv.org/pdf/1910.10683.pdf

[14] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 7871–7880. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.703

[15] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu, "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization," in *Proceedings of the Thirty-seventh International Conference on Machine Learning (IMCL)*, 2020.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "https://arxiv.org/abs/1706.03762," in *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS)*, 2017. [Online]. Available: https://arxiv.org/abs/1706.03762

[17] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," in *EMNLP*, 2019.

[18] S. Gehrmann, Y. Deng, and A. Rush, "Bottom-up abstractive summarization," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[19] M. T. Nayeem, T. A. Fuad, and Y. Chali, "Abstractive unsupervised multi-document summarization using paraphrastic sentence fusion," in *Proceedings of the International Conference on Computational Linguistics (COLING)*, 2018. [Online]. Available: https://www.aclweb.org/anthology/C18-1102

[20] Z. Cao, F. Wei, W. Li, and S. Li, "Faithful to the original: Fact aware neural abstractive summarization," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[21] T. Falke, L. F. R. Ribeiro, P. A. Utama, I. Dagan, and I. Gurevych, "Ranking generated summaries by correctness: An interesting but challenging application for natural language inference," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019. [Online]. Available: https://www.aclweb.org/anthology/P19-1213

[22] W. Kryściński, R. Paulus, C. Xiong, and R. Socher, "Improving abstraction in text summarization," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[23] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters, "The icsi meeting corpus," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, vol. 1, 2003, pp. I–I.

[24] I. Mccowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska Masson, W. Post, D. Reidsma, and P. Wellner, "The ami meeting corpus," *Int'l. Conf. on Methods and Techniques in Behavioral Research*, 01 2005.

[25] W. Kryściński, N. S. Keskar, B. McCann, C. Xiong, and R. Socher, "Neural text summarization: A critical evaluation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019. [Online]. Available: https://www.aclweb.org/anthology/D19-1051

[26] L. Lebanoff, J. Muchovej, F. Dernoncourt, D. S. Kim, S. Kim, W. Chang, and F. Liu, "Analyzing sentence fusion in abstractive summarization," in *Proceedings fo the EMNLP 2019 Workshop on New Frontiers in Summarization*, 2019. [Online]. Available: https://www.aclweb.org/anthology/D19-5413

[27] C. Li, F. Liu, F. Weng, and Y. Liu, "Document summarization via guided sentence compression," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013.

[28] K. Thadani and K. McKeown, "Sentence compression with joint structural inference," in *Proceedings of CoNLL*, 2013.

[29] L. Wang, H. Raghavan, V. Castelli, R. Florian, and C. Cardie, "A sentence compression based framework to query-focused multi-document summarization," in *Proceedings of ACL*, 2013.

[30] D. Yogatama, F. Liu, and N. A. Smith, "Extractive summarization by maximizing semantic volume," in *Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP)*, 2015.

[31] K. Filippova, E. Alfonseca, C. Colmenares, L. Kaiser, and O. Vinyals, "Sentence compression by deletion with lstms," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015.

[32] G. Durrett, T. Berg-Kirkpatrick, and D. Klein, "Learning-based single-document summarization with compression and anaphoricity constraints," in *Proceedings of the Association for Computational Linguistics (ACL)*, 2016.

[33] J. Cheng and M. Lapata, "Neural summarization by extracting sentences and words," in *Proceedings of ACL*, 2016.

[34] Z. Cao, W. Li, S. Li, and F. Wei, "Improving multi-document summarization via text classification," in *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*, 2017.

[35] M. Isonuma, T. Fujino, J. Mori, Y. Matsuo, and I. Sakata, "Extractive summarization using multi-task learning with document classification," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017.

[36] M. Yasunaga, R. Zhang, K. Meelu, A. Pareek, K. Srinivasan, and D. Radev, "Graph-based neural multi-document summarization," in *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)*, 2017.

[37] S. Narayan, S. B. Cohen, and M. Lapata, "Ranking sentences for extractive summarization with reinforcement learning," in *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2018.

[38] Y. Kikuchi, G. Neubig, R. Sasano, H. Takamura, and M. Okumura, "Controlling output length in neural encoder-decoders," in *Proceedings of EMNLP*, 2016.

[39] Q. Chen, X. Zhu, Z.-H. Ling, S. Wei, and H. Jiang, "Distraction-based neural networks for document summarization," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.

[40] Y. Miao and P. Blunsom, "Language as a latent variable: Discrete generative models for sentence compression," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016.

[41] W. Zeng, W. Luo, S. Fidler, and R. Urtasun, "Efficient summarization with read-again and copy mechanism," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[42] Q. Zhou, N. Yang, F. Wei, and M. Zhou, "Selective encoding for abstractive sentence summarization," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017.

[43] S. Narayan, S. B. Cohen, and M. Lapata, "Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018. [Online]. Available: https://arxiv.org/abs/1808.08745

[44] C. Gulcehre, S. Ahn, R. Nallapati, B. Zhou, and Y. Bengio, "Pointing the unknown words," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016.

[45] J. Gu, Z. Lu, H. Li, and V. O. Li, "Incorporating copying mechanism in sequence-to-sequence learning," in *Proceedings of ACL*, 2016.

[46] Y.-C. Chen and M. Bansal, "Fast abstractive summarization with reinforce-selected sentence rewriting," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.

[47] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *arXiv:1810.04805*, 2018. [Online]. Available: https://arxiv.org/abs/1810.04805

[48] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019. [Online]. Available: https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

[49] W.-T. Hsu, C.-K. Lin, M.-Y. Lee, K. Min, J. Tang, and M. Sun, "A unified model for extractive and abstractive summarization using inconsistency loss," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018. [Online]. Available: https://arxiv.org/abs/1805.06266

[50] Y. Liu and M. Lapata, "Hierarchical transformers for multi-document summarization," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 5070–5081. [Online]. Available: https://www.aclweb.org/anthology/P19-1500

[51] G. Carenini, R. Ng, and A. Pauls, "Multi-document summarization of evaluative text," in *Proceedings of 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2006.

[52] K.-F. Wong, M. Wu, and W. Li, "Extractive summarization using supervised and semi-supervised learning," in *Proceedings of the International Conference on Computational Linguistics (COLING)*, 2008.

[53] D. Galanis, G. Lampouras, and I. Androutsopoulos, "Extractive multi-document summarization with integer linear programming and support vector regression," in

Proceedings of the International Conference on Computational Linguistics (COLING), 2012. [Online]. Available: https://www.aclweb.org/anthology/C12-1056

[54] P. Li, W. Lam, L. Bing, and Z. Wang, "Deep recurrent generative decoder for abstractive text summarization," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017.

[55] K. Song, L. Zhao, and F. Liu, "Structure-infused copy mechanisms for abstractive summarization," in *Proceedings of the International Conference on Computational Linguistics (COLING)*, 2018.

[56] H. Guo, R. Pasunuru, and M. Bansal, "Soft, layer-specific multi-task summarization with entailment and question generation," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.

[57] Z. Cao, W. Li, S. Li, and F. Wei, "Retrieve, rerank and rewrite: Soft template based neural summarization," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018. [Online]. Available: https://aclweb.org/anthology/P18-1015

[58] C. Li, Y. Liu, F. Liu, L. Zhao, and F. Weng, "Improving multi-document summarization by sentence compression based on expanded constituent parse tree," in *Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP)*, 2014.

[59] B. Jing, Z. Wang, and E. Xing, "Show, describe and conclude: On exploiting the structure information of chest X-ray reports," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 6570–6580. [Online]. Available: https://www.aclweb.org/anthology/P19-1657

[60] Y. Zhang, D. Merck, E. B. Tsai, C. D. Manning, and C. P. Langlotz, "Optimizing the factual correctness of a summary: A study of summarizing radiology reports," in *Proceedings of the 58th Annual Conference of the Association for Computational Linguistics (ACL)*, 2020. [Online]. Available: https://arxiv.org/abs/1911.02541

[61] A. Kornilova and V. Eidelman, "BillSum: A corpus for automatic summarization of US legislation," in *Proceedings of the 2nd Workshop on New Frontiers in Summarization*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 48–56. [Online]. Available: https://www.aclweb.org/anthology/D19-5406

[62] Y. Mehdad, G. Carenini, F. W. Tompa, and R. T. NG, "Abstractive meeting summarization with entailment and fusion," in *Proceedings of the 14th European Workshop on Natural Language Generation*, 2013. [Online]. Available: https://www.aclweb.org/anthology/W13-2117

[63] M. Li, L. Zhang, H. Ji, and R. J. Radke, "Keep meeting summaries on topic: Abstractive multi-modal meeting summarization," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

[64] J. J. Koay, A. Roustai, X. Dai, A. Dillon, and F. Liu, "How domain terminology affects meeting summarization performance," in *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, 2020.

[65] M. Geva, E. Malmi, I. Szpektor, and J. Berant, "DISCOFUSE: A large-scale dataset for discourse-based sentence fusion," in *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019. [Online]. Available: https://www.aclweb.org/anthology/N19-1348

[66] L. Lebanoff, J. Muchovej, F. Dernoncourt, D. S. Kim, L. Wang, W. Chang, and F. Liu, "Understanding points of correspondence between sentences for abstractive summarization," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*. Seattle, United States: Association for Computational Linguistics, Jul. 2020. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-srw.26.pdf

[67] R. Barzilay, K. R. McKeown, and M. Elhadad, "Information fusion in the context of multi-document summarization," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 1999.

[68] G. Carenini and J. C. K. Cheung, "Extractive vs. NLG-based abstractive summarization of evaluative text: The effect of corpus controversiality," in *Proceedings of the Fifth International Natural Language Generation Conference (INLG)*, 2008.

[69] K. Ganesan, C. Zhai, and J. Han, "Opinosis: A graph-based approach to abstractive summarization of highly redundant opinions," in *Proceedings of the International Conference on Computational Linguistics (COLING)*, 2010.

[70] S. Gerani, Y. Mehdad, G. Carenini, R. T. Ng, and B. Nejat, "Abstractive summarization of product reviews using discourse structure," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[71] G. D. Fabbrizio, A. J. Stent, and R. Gaizauskas, "A hybrid approach to multi-document summarization of opinions in reviews," *Proceedings of the 8th International Natural Language Generation Conference (INLG)*, 2014.

[72] D. Pighin, M. Cornolti, E. Alfonseca, and K. Filippova, "Modelling events through memory-based, open-ie patterns for abstractive summarization," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2014.

[73] L. Bing, P. Li, Y. Liao, W. Lam, W. Guo, and R. J. Passonneau, "Abstractive multi-document summarization via phrase selection and merging," in *Proceedings of ACL*, 2015.

[74] F. Liu, J. Flanigan, S. Thomson, N. Sadeh, and N. A. Smith, "Toward abstractive summarization using semantic representations," in *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2015.

[75] K. Liao, L. Lebanoff, and F. Liu, "Abstract meaning representation for multi-document summarization," in *Proceedings of the International Conference on Computational Linguistics (COLING)*, 2018.

[76] R. Barzilay and K. R. McKeown, "Sentence fusion for multidocument news summarization," *Computational Linguistics*, vol. 31, no. 3, 2005. [Online]. Available: https://www.mitpressjournals.org/doi/pdf/10.1162/089120105774321091

[77] E. Marsi and E. Krahmer, "Explorations in sentence fusion," in *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, 2005. [Online]. Available: https://www.aclweb.org/anthology/W05-0701

[78] K. Filippova and M. Strube, "Sentence fusion via dependency graph compression," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2008. [Online]. Available: https://www.aclweb.org/anthology/D08-1019

[79] J. C. K. Cheung and G. Penn, "Unsupervised sentence enhancement for automatic summarization," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[80] R. Barzilay and K. R. McKeown, "Sentence Fusion for Multidocument News Summarization," *Computational Linguistics*, vol. 31, no. 3, pp. 297–328, Sep. 2005. [Online]. Available: http://www.mitpressjournals.org/doi/10.1162/089120105774321091

[81] K. Filippova, "Multi-sentence compression: Finding shortest paths in word graphs," in *Proceedings of the International Conference on Computational Linguistics (COLING)*, 2010. [Online]. Available: https://www.aclweb.org/anthology/C10-1037

[82] C. Shen and T. Li, "Multi-document summarization via the minimum dominating set," in *Proceedings of the International Conference on Computational Linguistics (COLING)*, 2010. [Online]. Available: https://www.aclweb.org/anthology/C10-1111

[83] V. Chenal and J. C. K. Cheung, "Predicting sentential semantic compatibility for aggregation in text-to-text generation," in *Proceedings of the International Conference on Computational Linguistics (COLING)*, 2016. [Online]. Available: https://www.aclweb.org/anthology/C16-1101

[84] A. F. T. Martins and N. A. Smith, "Summarization with a joint model for sentence extraction and compression," in *Proceedings of the ACL Workshop on Integer Linear Programming for Natural Language Processing*, 2009.

[85] H. Li, J. Zhu, J. Zhang, and C. Zong, "Ensure the correctness of the summary: Incorporate entailment knowledge into abstractive sentence summarization," in *Proceedings of the International Conference on Computational Linguistics (COLING)*, 2018.

[86] J. W. G. Putra, H. Kobayashi, and N. Shimizu, "Incorporating topic sentence on neural news headline generation," 2018. [Online]. Available: https://pdfs.semanticscholar.org/8c72/cab320ce50330e8e45d5bf79f635bb724500.pdf

[87] K. McKeown, S. Rosenthal, K. Thadani, and C. Moore, "Time-efficient creation of an accurate sentence fusion corpus," in *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2010. [Online]. Available: https://www.aclweb.org/anthology/N10-1044

[88] M. Elsner and D. Santhanam, "Learning to fuse disparate sentences," in *Proceedings of ACL Workshop on Monolingual Text-To-Text Generation*, 2011.

[89] K. Thadani and K. McKeown, "Supervised sentence fusion with single-stage inference," in *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)*, 2013. [Online]. Available: https://www.aclweb.org/anthology/I13-1198

[90] A. Moryossef, Y. Goldberg, and I. Dagan, "Step-by-Step: Separating planning from realization in neural data-to-text generation," in *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019. [Online]. Available: https://www.aclweb.org/anthology/N19-1236

[91] W. Zhao, M. Peyrard, F. Liu, Y. Gao, C. M. Meyer, and S. Eger, "MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance," in *Proc. of EMNLP*, 2019.

[92] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=SkeHuCVFDr

[93] E. Durmus, H. He, and M. Diab, "Feqa: A question answering evaluation framework for faithfulness assessment in abstractive summarization," in *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*, 2020. [Online]. Available: https://arxiv.org/abs/2005.03754

[94] A. Wang, K. Cho, and M. Lewis, "Asking and answering questions to evaluate the factual consistency of summaries," in *Proceedings of the Annual Conference of the Association for Computational Linguistics (ACL)*, 2020. [Online]. Available: https://arxiv.org/abs/2004.04228

[95] M. A. K. Halliday and R. Hasan, *Cohesion in English*. English Language Series. Longman Group Ltd., 1976.

[96] A. Gatt and E. Krahmer, "Survey of the state of the art in natural language generation: Core tasks, applications and evaluation," *Journal of Artificial Intelligence Research*, vol. 61, pp. 65–170, 2018.

[97] C.-Y. Lin, "ROUGE: a package for automatic evaluation of summaries," in *Proceedings of ACL Workshop on Text Summarization Branches Out*, 2004.

[98] T. Baumel, M. Eyal, and M. Elhadad, "Query focused abstractive summarization: Incorporating query relevance, multi-document coverage, and summary length constraints into seq2seq models," *arXiv preprint arXiv:1801.07704*, 2018.

[99] J. Zhang, J. Tan, and X. Wan, "Towards a neural network approach to abstractive multi-document summarization," *arXiv preprint arXiv:1804.09010*, 2018.

[100] J. Carbonell and J. Goldstein, "The use of MMR, diversity-based reranking for reordering documents and producing summaries," in *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 1998.

[101] A. Nenkova and K. McKeown, "Automatic summarization," *Foundations and Trends in Information Retrieval*, 2011.

[102] A. Haghighi and L. Vanderwende, "Exploring content models for multi-document summarization," in *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2009.

[103] A. K. B. Taskar, *Determinantal Point Processes for Machine Learning*. Now Publishers Inc., 2012.

[104] A. Fan, D. Grangier, and M. Auli, "Controllable abstractive summarization," in *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, 2018.

[105] P. Laban, A. Hsi, J. Canny, and M. A. Hearst, "The summary loop: Learning to write abstractive summaries without examples," in *Proceedings of ACL*, Jul. 2020.

[106] L. Perez-Beltrachini, Y. Liu, and M. Lapata, "Generating summaries with topic templates and structured convolutional decoders," in *Proceedings of ACL*, 2019.

[107] W. Li, X. Xiao, J. Liu, H. Wu, H. Wang, and J. Du, "Leveraging graph to improve abstractive multi-document summarization," in *Proceedings of ACL*, 2020.

[108] A. Bražinskas, M. Lapata, and I. Titov, "Few-shot learning for abstractive multi-document opinion summarization," 2020.

[109] M. Coavoux, H. Elsahar, and M. Gallé, "Unsupervised aspect-based multi-document abstractive summarization," in *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, 2019.

[110] M. Grenander, Y. Dong, J. C. K. Cheung, and A. Louis, "Countering the effects of lead bias in news summarization via multi-stage training and auxiliary losses," in *Proceedings of EMNLP*, Nov. 2019.

[111] H. D. III and D. Marcu, "Generic sentence fusion is an ill-defined summarization task," in *Proceedings of ACL Workshop on Text Summarization Branches Out*, 2004.

[112] P. Over and J. Yen, "An introduction to DUC-2004," *National Institute of Standards and Technology*, 2004.

[113] H. T. Dang and K. Owczarzak, "Overview of the TAC 2008 update summarization task," in *Proceedings of Text Analysis Conference (TAC)*, 2008.

[114] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *Proceedings of Neural Information Processing Systems (NIPS)*, 2015.

[115] M. Grusky, M. Naaman, and Y. Artzi, "Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies," in *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018.

[116] A. Cohan, F. Dernoncourt, D. S. Kim, T. Bui, S. Kim, W. Chang, and N. Goharian, "A discourse-aware attention model for abstractive summarization of long documents," in *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018. [Online]. Available: https://arxiv.org/abs/1804.05685

[117] E. Sharma, L. Huang, Z. Hu, and L. Wang, "An entity-driven framework for abstractive summarization," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

[118] M. Koupaee and W. Y. Wang, "Wikihow: A large scale text summarization dataset," *arXiv preprint arXiv:1810.09305*, 2018.

[119] B. Kim, H. Kim, and G. Kim, "Abstractive Summarization of Reddit Posts with Multi-level Memory Networks," Nov. 2018. [Online]. Available: https://arxiv.org/abs/1811.00783

[120] R. Zhang and J. Tetreault, "This email could save your life: Introducing the task of email subject line generation," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 446–456. [Online]. Available: https://www.aclweb.org/anthology/P19-1043

[121] P. J. Liu, M. Saleh, E. Pot, B. Goodrich, R. Sepassi, L. Kaiser, and N. Shazeer, "Generating wikipedia by summarizing long sequences," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018. [Online]. Available: https://arxiv.org/abs/1801.10198

[122] A. Fabbri, I. Li, T. She, S. Li, and D. Radev, "Multi-News: A large-scale multi-document summarization dataset and abstractive hierarchical model," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

[123] D. Gholipour Ghalandari, C. Hokamp, N. T. Pham, J. Glover, and G. Ifrim, "A large-scale multi-document summarization dataset from the Wikipedia current events portal," in *Proceedings of ACL*, 2020.

[124] J.-P. Ng and V. Abrecht, "Better summarization evaluation with word embeddings for ROUGE," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 1925–1930. [Online]. Available: https://www.aclweb.org/anthology/D15-1222

[125] K. Ganesan, "Rouge 2.0: Updated and improved measures for evaluation of summarization tasks," *arXiv preprint arXiv:1803.01937*, 2018.

[126] E. ShafieiBavani, M. Ebrahimi, R. Wong, and F. Chen, "A graph-theoretic summary evaluation for ROUGE," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 762–767. [Online]. Available: https://www.aclweb.org/anthology/D18-1085

[127] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating text generation with BERT," in *https://arxiv.org/abs/1904.09675*, 2019.

[128] W. Zhao, M. Peyrard, F. Liu, Y. Gao, C. M. Meyer, and S. Eger, "MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

[129] T. Scialom, S. Lamprier, B. Piwowarski, and J. Staiano, "Answers unite! unsupervised metrics for reinforced summarization models," 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)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3246–3256. [Online]. Available: https://www.aclweb.org/anthology/D19-1320

[130] A. Kulesza and B. Taskar, "Learning determinantal point processes," in *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2011. [Online]. Available: https://dl.acm.org/citation.cfm?id=3020597

[131] S. Cho, L. Lebanoff, H. Foroosh, and F. Liu, "Improving the similarity measure of determinantal point processes for extractive multi-document summarization," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

[132] C. Kedzie, K. McKeown, and H. D. III, "Content selection in deep learning models of summarization," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018. [Online]. Available: https://arxiv.org/abs/1810.12343

[133] L. Lebanoff, K. Song, and F. Liu, "Adapting the neural encoder-decoder framework from single to multi-document summarization," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018. [Online]. Available: https://aclweb.org/anthology/D18-1446

[134] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., 2017.

[135] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[136] K. Hong and A. Nenkova, "Improving the estimation of word importance for news multi-document summarization," in *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2014. [Online]. Available: https://www.aclweb.org/anthology/E14-1075

[137] D. Chen, J. Bolton, and C. D. Manning, "A thorough examination of the cnn/daily mail reading comprehension task," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016.

[138] L. Vanderwende, H. Suzuki, C. Brockett, and A. Nenkova, "Beyond SumBasic: Task-focused summarization with sentence simplification and lexical expansion," *Information Processing and Management*, vol. 43, no. 6, pp. 1606–1618, 2007.

[139] G. Erkan and D. R. Radev, "LexRank: Graph-based lexical centrality as salience in text summarization," *Journal of Artificial Intelligence Research*, 2004.

[140] L. Lebanoff, K. Song, F. Dernoncourt, D. S. Kim, S. Kim, W. Chang, and F. Liu, "Scoring sentence singletons and pairs for abstractive summarization," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

[141] H. Jing, "Sentence reduction for automatic text summarization," in *Proceedings of the Sixth Conference on Applied Natural Language Processing*, 2000.

[142] L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider, "Abstract meaning representation for sembanking," in *Proceedings of Linguistic Annotation Workshop*, 2013.

[143] E. Reiter, "A structured review of the validity of BLEU," *Computational Linguistics*, vol. 44, no. 3, pp. 393–401, 2018.

[144] C. Li, X. Qian, and Y. Liu, "Using supervised bigram-based ILP for extractive summarization," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2013.

[145] R. McDonald, "Discriminative sentence compression with soft syntactic evidence," in *Proceedings of EACL*, 2006.

[146] J. Novikova, O. Dušek, A. Cercas Curry, and V. Rieser, "Why we need new evaluation metrics for NLG," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 2241–2252. [Online]. Available: https://www.aclweb.org/anthology/D17-1238

[147] V. Ng, "Machine learning for entity coreference resolution: A retrospective look at two decades of research." in *Proceedngs of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

[148] J. Lu and V. Ng, "Event coreference resolution: A survey of two decades of research," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

[149] H. Hardy, S. Narayan, and A. Vlachos, "HighRES: Highlight-based reference-less evaluation of summarization," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

[150] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The stanford CoreNLP natural language processing toolkit," in *Proceedings of the Association for Computational Linguistics (ACL) System Demonstrations*, 2014.

[151] M. Honnibal and I. Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," 2017, to appear.

[152] M. Gardner, J. Grus, M. Neumann, O. Tafjord, P. Dasigi, N. F. Liu, M. Peters, M. Schmitz, and L. S. Zettlemoyer, "AllenNLP: A deep semantic natural language processing platform," *arXiv:1803.07640*, 2017.

[153] A. Bagga and B. Baldwin, "Algorithms for scoring coreference chains," in *The first international conference on language resources and evaluation workshop on linguistics coreference*, vol. 1. Granada, 1998, pp. 563–566. [Online]. Available: https://pdfs.semanticscholar.org/4b51/2f10838e05f5b2eee94bfbd20f3d9c4ecb9b.pdf

[154] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, and H.-W. Hon, "Unified language model pre-training for natural language understanding and generation," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, Eds., 2019.

[155] E. Strubell, P. Verga, D. Andor, D. Weiss, and A. McCallum, "Linguistically-informed self-attention for semantic role labeling," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[156] B. J. Grosz, A. K. Joshi, and S. Weinstein, "Centering: A framework for modeling the local coherence of discourse," *Comp. Ling.*, 1995.

[157] K. Papineni, S. Roukos, T. Ward, , and W. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002.

[158] F. Ladhak, B. Li, Y. Al-Onaizan, and K. McKeown, "Exploring content selection in summarization of novel chapters," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 5043–5054. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.453

[159] S. Takase, J. Suzuki, N. Okazaki, T. Hirao, and M. Nagata, "Neural headline generation on abstract meaning representation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016.

[160] E. Sandhaus, "The new york times annotated corpus," *Linguistic Data Consortium*, 2008.

[161] W. Luo and D. Litman, "Summarizing student responses to reflection prompts," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015.

[162] W. Luo, F. Liu, Z. Liu, and D. Litman, "Automatic summarization of student course feedback," in *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2016.

[163] K. Hong, J. M. Conroy, B. Favre, A. Kulesza, H. Lin, and A. Nenkova, "A repository of state of the art and competitive baseline summaries for generic news summarization," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC)*, 2014.

[164] H. Jing and K. McKeown, "The decomposition of human-written summary sentences," in *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 1999.

[165] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014.

[166] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[167] D. Gillick, B. Favre, D. Hakkani-Tur, B. Bohnet, Y. Liu, and S. Xie, "The ICSI/UTD summarization system at TAC 2009," in *Proceedings of TAC*, 2009.

[168] R. K. Amplayo and M. Lapata, "Unsupervised opinion summarization with noising and denoising," in *Proceedings of ACL*, 2020.

[169] J. Hamilton. (2014, October) A day in the life of an intelligence analyst. [Online]. Available: https://news.clearancejobs.com/2014/10/30/day-life-intelligence-analyst/

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

[171] D. L. Hintzman, "Repetition and memory," *Psychology of Learning and Motivation*, vol. 10, pp. 47–91, 1976.

[172] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020.

[173] A. Handler and B. O'Connor, "Relational summarization for corpus analysis," in *Proceedings of NAACL*, Jun. 2018.

[174] W. Zhao, M. Peyrard, F. Liu, Y. Gao, C. M. Meyer, and S. Eger, "MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 563–578. [Online]. Available: https://www.aclweb.org/anthology/D19-1053

[175] R. Mihalcea and P. Tarau, "TextRank: Bringing order into text," in *Proceedings of EMNLP*, 2004.

[176] H. Lin and J. Bilmes, "A class of submodular functions for document summarization," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, Jun. 2011, pp. 510–520. [Online]. Available: https://www.aclweb.org/anthology/P11-1052

[177] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[178] A. Nenkova and R. Passonneau, "Evaluating content selection in summarization: The pyramid method," in *Proceedings of NAACL*, 2004.

[179] A. Bražinskas, M. Lapata, and I. Titov, "Unsupervised opinion summarization as copycat-review generation," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 5151–5169. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.461