

The
University
Of
Sheffield.

Guiding Abstractive Summarization using Structural Information

by
Hardy

A thesis submitted in partial fulfilment of the requirements for the degree
of
Doctor of Philosophy

The University of Sheffield
Faculty of Engineering
Department of Computer Science

September 2021

Declaration

I, the author, confirm that the Thesis is my own work. I am aware of the University's Guidance on the Use of Unfair Means (www.sheffield.ac.uk/ssid/unfair-means). This work has not been previously been presented for an award at this, or any other, university.

Hardy

28 August 2021

Acknowledgements

I would like to thank my wife and aunt for everything that they have given me, my supervisor Andreas Vlachos and Nikolaos Aletras for their help and guidance during my PhD, my collaborator Shashi Narayan, my friends in the NLP lab of University of Sheffield and lastly the Indonesian government that has sponsored my studies through the Indonesia Endowment Fund for Education (LPDP).

Abstract

Abstractive summarization takes a set of sentences from a source document and reproduces its salient information using the summarizer's own words into a summary. Produced summaries may contain novel words and have different grammatical structures from the source document. In a sense, abstractive summarization is closer to how a human summarizes, yet it is also more difficult to automate since it requires a full understanding of the natural language. However, with the inception of deep learning, many new summarization systems achieved improved automatic and manual evaluation scores. One prominent deep learning model is the sequence-to-sequence model with an attention-based mechanism. Moreover, the advent of pre-trained language models over a huge set of unlabeled data further improved the performance of a summarization system. However, with all the said improvements, abstractive summarization is still adversely affected by hallucination and disfluency. Furthermore, all these recent works that used a seq2seq model require a large dataset since the underlying neural network easily overfits on a small dataset resulting in a poor approximation and high variance outputs. The problem is that these large datasets often came with only a single reference summary for each source document despite that

it is known that human annotators are subject to a certain degree of subjectivity when writing a summary.

We addressed the first problem by using a mechanism where the model uses a guidance signal to control what tokens are to be generated. A guidance signal can be defined as different types of signals that are fed into the model in addition to the source document where a commonly used one is structural information from the source document. Recent approaches showed good results using this approach, however, they were using a joint-training approach for the guiding mechanism, in other words, the model needs to be re-trained if a different guidance signal is used which is costly. We propose approaches that work without re-training and therefore are more flexible with regards to the guidance signal source and also computationally cheaper. We performed two different experiments where the first one is a novel guided mechanism that extends previous work on abstractive summarization using Abstract Meaning Representation (AMR) with a neural language generation stage which we guide using side information. Results showed that our approach improves over a strong baseline by 2 ROUGE-2 points. The second experiment is a guided key-phrase extractor for more informative summarization. This experiment showed mixed results, but we provide an analysis of the negative and positive output examples.

The second problem was addressed by our proposed manual evaluation framework called HIGHLIGHT-based Reference-less Evaluation Summarization (HighRES). The proposed framework avoids reference bias and provides absolute instead of ranked evaluation of the systems. To validate our approach we employed crowd-workers to augment with highlights on the eXtreme SUMmarization (XSUM) dataset which is a highly abstractive

summarization dataset. We then compared two abstractive systems (Pointer Generator and T-Conv) to demonstrate our approach. Results showed that HighRES improves inter-annotator agreement in comparison to using the source document directly, while it also emphasizes differences among systems that would be ignored under other evaluation approaches. Our work also produces annotated dataset which gives more understanding on how humans select salient information from the source document.

Contents

Declaration	i
Acknowledgements	ii
Abstract	iii
List of Figures	ix
List of Tables	xi
1 Introduction	1
1.1 Contributions and Publications	6
1.2 Report Structure	8
2 Background	10
2.1 Deep Learning (DL)	10
2.1.1 Recurrent Neural Networks (RNNs)	11
2.1.2 Convolutional Neural Networks (CNNs)	15
2.1.3 Transformer Networks	17
2.2 Pre-trained Language Model	19
2.2.1 Bidirectional Encoder Representations from Transformers (BERT)	20
2.2.2 BART	21
2.2.3 Text-to-Text Transfer Transformer (T5) approach	22
2.2.4 Pegasus	23
2.3 Automatic Summarization	23
2.3.1 Multi-Document and Single Document Summarization	24
2.3.2 Extractive and Abstractive Summarization	27
2.3.3 Summarization using Structural Information	31
2.3.4 Summarization Using a Guiding Mechanism	37
2.4 Guidance by Constraining Beam Search	41
2.4.1 Usage in Neural Machine Translation (NMT)	42
2.4.2 Lexically Constrained Decoding	44
2.5 Automatic and Manual Evaluation in Summarization	46

2.5.1	Common Criteria for Summarization	46
2.5.2	Automatic Evaluation	48
2.5.3	Manual Evaluation	51
2.6	Datasets for Summarization	61
2.7	Conclusion and Final Remarks	63
3	HighRES: Highlight-based Reference-less Evaluation of Summarization	67
3.1	Introduction	68
3.2	HIGHRES	69
3.2.1	Highlight Annotation	71
3.2.2	Highlight-based Content Evaluation	72
3.2.3	Clarity and Fluency Evaluations	73
3.2.4	Highlight-based ROUGE Evaluation	76
3.3	Summarization Dataset and Models	78
3.4	Experiments	79
3.4.1	Highlight Annotation	79
3.4.2	Content Evaluation of Summaries	82
3.4.3	Clarity and Fluency Evaluation	84
3.4.4	Highlight-based ROUGE Evaluation	84
3.5	Qualitative Analysis	85
3.5.1	HIGHRES eliminates reference bias.	85
3.5.2	Fluency vs Clarity.	86
3.6	Conclusion	87
4	Guided Neural Language Generation of AMR for Summarization	89
4.1	Introduction	90
4.2	Methodology	92
4.2.1	AMR-based summarization	92
4.2.2	Unguided NLG from AMR	93
4.2.3	Guided NLG from AMR	94
4.3	Experiments	97
4.4	Conclusion and Future Works	102
5	Guided Key-phrase Extraction for More Informative Summarization	105
5.1	Introduction	106
5.2	Methodology	108
5.2.1	Summarizer model training	108
5.2.2	GKESUM extraction training	109
5.2.3	Summarizer model inference	113
5.3	Experimental Setup and Result	113
5.3.1	Data	113
5.3.2	Model	113

5.3.3	Evaluation for the Key-phrase Extractor Model	114
5.3.4	Automatic Evaluation for the Summarizer Model	114
5.3.5	Human Evaluation	116
5.3.6	Qualitative Analysis	117
5.4	Conclusion and Future Work	119
6	Conclusion and Future Work	120
6.1	Summary of Findings	122
6.2	Future Work	123
A	DUC 2003 Summary Quality Questions	124
B	DUC2004 Summary Quality Question	126
C	DUC2005 Summary Quality Question	130
C.1	Readability Task	130
C.2	Responsiveness Task	134
	Bibliography	136

List of Figures

2.1	BERT illustration (Devlin et al., 2019).	20
2.2	A compression of two sentences using dependency trees. Dropped words are colored in a lighter color. (Berg-Kirkpatrick et al., 2011).	29
2.3	An AMR Graph of the sentence ‘the boy wants the girl to believe him.’ (Liu et al., 2015) There is a re-entrancy in the concept node of ‘boy’ which has several parents.	35
2.4	A joined AMR graph from two sentences AMR graphs. (1) and (2) are added edges through expansion steps. (Liu et al., 2015)	36
2.5	Different guidance signals and its corresponding output using Dou et al. (2020)’s model	41
2.6	The Summary Evaluation Environment 2.0 user interface. (Lin, 2001)	53
3.1	Highlight-based evaluation of summaries. Annotators to evaluate a summary (bottom) against the highlighted source document (top) presented with a heat map marking the salient content in the document; the darker the colour, the more annotators deemed the highlighted text salient.	68
3.2	The UI for highlight annotation. Judges are given an article and asked to highlight words or phrases that are important in the article.	71
3.3	The sanity checking question at the end of the annotation task.	72
3.4	The UI for content evaluation with highlights. Judges are given an article with important words highlighted using a heat map. Judges can also remove less important highlight color by sliding the scroller at the left of the page. At the right of the page, judges give the recall and precision assessment by sliding the scroller from 1 to 100 based on the given summary quality.	73
3.5	The UI for content evaluation without highlight. At the right of the page, judges give the recall and precision assessment by sliding the scroller from 1 to 100 based on the given summary quality.	74
3.6	The UI for content evaluation using a reference summary as a comparison. At the right of the page, judges give the recall and precision assessment by sliding the scroller from 1 to 100 based on the given summary quality.	75
3.7	The UI for fluency evaluation. Judges are given a number of summaries which can be switched by pressing the ‘Prev’ or ‘Next’ button. To give an assessment, there is a scroller from 1 to 100.	75

3.8	The UI for clarity evaluation. Judges are given a number of summaries which can be switched by pressing the ‘Prev’ or ‘Next’ button. To give an assessment, there is a scroller from 1 to 100.	76
3.9	Highlight annotation for the documents with the highest (left) and lowest (right) agreement. We also show their reference summaries at the bottom.	80
3.10	The highlighted article, reference summary, and summaries are generated by TCONVS2S and PTGEN. Words in red in the system summaries are highlighted in the article but do not appear in the reference.	86
4.1	An Overview Diagram of the Guided NLG of AMR for summarization. .	89
4.2	An example of the original AMR (left) and the variable-free AMR (right) displaying the meaning of <i>Opium is the raw material used to make heroin</i> (Van Noord and Bos, 2017).	94
4.3	Effect of different equations for combining the probability distribution of the decoder with the side information.	96
5.1	An overview diagram of the guided key-phrase extraction for more informative summarization.	105

List of Tables

2.1	Comparison of Li et al. (2018b) model’s output, the pointer generator baseline and the gold summary.	39
2.2	Overview of manual evaluations conducted in recent summarization systems. We categorize them in four dimensions: the first column presents papers that do not report on human evaluation; the second column identifies matrices used for evaluating content (“ <i>Pyramid</i> ”, “ <i>QA</i> ”, “ <i>Correctness</i> ”, “ <i>Recall</i> ” and “ <i>Precision</i> ”) and quality (“ <i>Clarity</i> ”, “ <i>Fluency</i> ”) of summaries; the third column focuses if the system ranking reported by humans on content evaluation were “ <i>Absolute</i> ” or “ <i>Relative</i> ”; and finally, the fourth column evaluates if summaries were evaluated against the input document (“ <i>With Document</i> ”), the reference summary (“ <i>With Reference</i> ”) or both (“ <i>With Ref. & Doc.</i> ”).	65
2.3	Various dataset for summarization purpose	66
3.1	Results of content evaluation of summaries against documents with highlights, documents without highlights and reference summaries.	81
3.2	Coefficient of variation (lower is better) for evaluating summaries against documents with and without highlights.	81
3.3	Mean ”Fluency” and ”Clarity” scores for TCONVS2S , PTGEN and Reference summaries. All the ratings were collected on a 1-100 Likert scale.	83
3.4	HROUGE-1 (unigram) and HROUGE-2 (bigram) precision, and recall scores for TCONVS2S , PTGEN and Reference summaries.	84
3.5	TCONVS2S and PTGEN showing a disagreement between fluency and clarity scores. We italicized words that are not clear in the summaries.	87
4.1	Neural network model hallucination.	90
4.2	System outputs of Liu et al. (2015). The bottom bag-of-words are generated from the top AMR tree.	93
4.3	Insights of different equations.	97
4.4	Results for AMR-to-text	98
4.5	BLEU and ROUGE results for guided and unguided models using test dataset.	99

4.6	The F_1 ROUGE scores for guided, unguided, Liu et al. (2015) (BoW) results in Gold and RIGA parses, and seq2seq summarization. All models are run using test dataset.	100
4.7	Result summaries of guided, unguided and seq2seq models compared with gold summary.	101
4.8	Fluency scores on test dataset.	101
4.9	Problems in guided model’s summaries.	102
5.1	GKE-SUM and BERTKPE performance using Recall, Precision and F1 scores on one, three and five key-phrases size	114
5.2	Results on three abstractive summarization datasets.	115
5.3	Results on XSUM dataset with different settings. The chronological is fine-tuned on the year prior to 2017 and tested on after 2017. The distilled is XSUM fine-tuned using the distilled version of BART	115
5.4	Oracle constraints results of BART-GKESUM on XSUM dataset.	115
5.5	Human evaluation results of BART and BART-GKESUM’s generated summaries on Reddit.	117
5.6	A positive sample shown in summaries of BART, BART-GKESUM and gold. Constraints provide missing information and makes BART-GKESUM’s summary more informative.	117
5.7	A negative sample shown in summaries of BART, BART-GKESUM and gold. Adding constraints that have already been satisfied by the summarization model causes hallucination in BART-GKESUM’s summary	118

Chapter 1

Introduction

Since the advent of the computer, written text format has been the most popular form of information. However, the rapid growth of textual information that comes with the Internet is overwhelming. For this reason, it is necessary to have an intelligent approach that is capable to process and summarize large amounts of information. Automatic summarization is the task where given a text an intelligent system can extract all salient information from it and compile it into an informative and fluent summary. There are many different types of automatic summarization in used today. We will discuss in detail different types of summarization in Chapter 2 but in this Thesis, we focus on one prominent type of summarization: abstractive summarization.

Based on the method of producing a summary, there are two types of summarization: extractive and abstractive summarization. Extractive summarization focuses on selecting salient sentences from the document and then presents them directly as a summary, thus

the result can sometimes lack coherence. Abstractive summarization, on the other hand, draws salient information from the document and paraphrases them into a summary. This characteristic improves the coherence of a summary but it is also more difficult to perform as it requires a full understanding of the natural language.

Today, with the inception of the deep learning approach, many new summarization systems (Rush et al., 2015; See et al., 2017; Paulus et al., 2018; Lewis et al., 2020; Zhang et al., 2020) that used it saw a major boost in their automatic and manual evaluation scores. One prominent deep learning architecture that enables the performance boost is the sequence-to-sequence (Sutskever et al., 2014, seq2seq) model with an attention-based mechanism (Bahdanau et al., 2015; Luong et al., 2015). The recent state-of-the-art approaches (Lewis et al., 2020; Zhang et al., 2020) of abstractive summarization are based on the seq2seq model (Bahdanau et al., 2015; Luong et al., 2015).

The seq2seq model falls into two neural networks: the encoder and decoder networks. The encoder encodes a sequence of source tokens into a single vector representation of which is later decoded by the decoder into a new sequence of target tokens. In a summarization context, this enables the neural model to capture information from the text and then generates a new and shorter sequence of words but contain only salient information.

Moreover, the advent of pre-trained language models (Radford et al., 2018; Peters et al., 2018; Devlin et al., 2019) over a huge set of unlabeled data further improves the performance of summarization systems (Liu and Lapata, 2019; Lewis et al., 2020; Zhang et al., 2020). This is because leveraging a pre-trained language model, helps the

neural model to develop general knowledge over many topics and improves the fluency of language generation. However, with all said improvements, abstractive summarization is still adversely affected by hallucination and disfluency. Hallucination is defined as the generation of tokens by the model of which are not relevant to the document. This definition is adapted from another work of language generation that is related to image captioning (Rohrbach et al., 2018). Disfluency simply means that the generated text does not conform the grammatical rules or unintelligible.

We addressed hallucination and disfluency problems by using a mechanism where the model uses a guidance signal to control what tokens are to be generated. A guidance signal can be defined as different types of signals that are fed into the model in addition to the source document where a commonly used one is structural information from the source document. A structural information is information that comes from the structure of the text itself, for examples: constituency, AMR, dependency structures and many others. Leveraging structural information can help language generation since it provides additional contexts to the model when generating a token.

There are many attempts of using guidance signals to aid summarization but we will focus on works that used a neural approach as the non-neural approach couldn't produce a fluent and coherent output. One example of such works was done by Li et al. (2018a) which used key-phrases extracted from the structure of the source document using TextRank(Mihalcea and Tarau, 2004) and then incorporated them in a neural network which they called Key Information Guide Network. Li et al. (2020) further improves their work by using key-phrases to improve the encoding representations in the encoder

network. Other such as [Dou et al. \(2020\)](#) used varieties of structural information such as highlighted sentences, relations, keywords and retrieved summaries as guides for their model. All these models, however, were using a joint-training approach for the guiding mechanism, in other words, to include a different source of information into the model it has to be re-trained which is costly.

We propose approaches that work without re-training and therefore are more flexible with regards to the guidance signal source and also computationally cheaper. We performed two different experiments where the first one is a novel guided mechanism that extends previous work on abstractive summarization using Abstract Meaning Representation (AMR) with a neural language generation stage which we guide using side information. Results showed that our approach improves over a strong baseline by 2 ROUGE-2 points. The second experiment is a guided key-phrase extractor for more informative summarization. This experiment showed mixed results, but we provide an analysis of the negative and positive output examples.

Another problem is that all recent works in abstractive summarization that use the seq2seq model, on the other hand, require a large dataset ([Hermann et al., 2015](#); [Narayan et al., 2018c](#); [Koupaei and Wang, 2018](#); [Kim et al., 2018](#)) since the underlying neural network is easily overfit on a small dataset resulting in a poor approximation and high variance outputs. The problem is that these large datasets often come with only a single reference the summary for each source document despite that it is known ([Harman and Over, 2004](#)) that a human annotator is subject to a certain degree of subjectivity when writing a summary. In other words there could be many possible good summaries. This

problem is called reference bias (Louis and Nenkova, 2013; Fomicheva and Specia, 2016). ROUGE which was designed for multiple summaries evaluation is also a poor choice for these datasets since they are heavily biased to the single reference summary. A manual evaluation is required to properly evaluate summary’s quality however this approach is also referenced biased when it is done by evaluating the difference between the system and reference summary.

We addressed the reference bias problem by proposing a new approach of manual evaluation (Chapter 3) that isn’t biased to the single reference summaries. The second problem was addressed by our proposed manual evaluation framework called HIGHLIGHT-based Reference-less Evaluation Summarization (HighRES). The proposed framework avoids reference bias and provides absolute instead of ranked evaluation of the systems. To validate our approach we employed crowd-workers to annotate the eXtreme SUMmarization (XSUM) dataset with highlights. We then compare two abstractive systems (Pointer Generator and T-Conv) to demonstrate our approach. Results showed that HighRES improves inter-annotator agreement in comparison to using the source document directly, while it also emphasizes differences among systems that would be ignored under other evaluation approaches. Our work also produces annotated dataset which gives more understanding of how humans select salient information from the source document. HighRES, however, couldn’t be applied to our other experiments since each experiment uses datasets that are different from the HighRES annotated dataset.

As such our research questions are as follows.

1. Can we devise a new manual evaluation system that is not affected by the reference bias problem?
2. How to better leverage structural information as a guidance signal to address hallucination and disfluency problems by an abstractive summarization system?
3. Can we incorporate structural information into the guiding mechanism without having to re-train the model every time we change the information source?

1.1 Contributions and Publications

Contributions of this Thesis span across three experiments that were conducted to answer our research questions. We list our contributions as follows.

1. A novel guidance mechanism ([Hardy and Vlachos, 2018](#)) in Chapter 4 that works by leveraging Abstractive Meaning Representation ([Banarescu et al., 2013](#), AMR) in the source document to increase fluency and informativeness of summaries. We based our work on an existing AMR summarization by [Liu et al. \(2015\)](#). Our approach can be applied without having to re-train the model. Results show that our approach improved over our chosen baseline ([See et al., 2017](#)) by 2 ROUGE-2 points. This contribution addresses the second and third research questions, however, our work has a limitation where due to small dataset it is difficult to show significance in the result therefore further experiment on larger data is needed to validate our approach. Nevertheless, we devise a new manual evaluation (HighRES)

in order to give a better perspective on our approach but due to time and cost constraints we did not manage to apply HighRES on this experiment.

2. A novel approach in manual evaluation named Highlight-based Reference-less Evaluation of Summarization (Hardy et al., 2019, HighRES) in Chapter 3. We devised a new approach that allows manual evaluation by using the highlighted source document without the need of a reference summary therefore eliminating the reference bias problem. This contribution addresses the third research question.
3. A new annotated dataset for HighRES evaluation (Hardy et al., 2019). We performed crowd-sourced annotation to obtain a highlighted source document for the purpose of HighRES evaluation. This annotated dataset also gives better understanding on how humans select salient information from the source document. Subsequently our work in Chapter 5 was inspired by this insight.
4. We devised a guided key-phrase extractor for more informative summarization (Chapter 5) however we couldn't gain a significant improvement despite a promising oracle result. We, however, perform analysis to gain insights on why it couldn't work and leave it for future works.

Our works are published in two major conferences papers. Chapter 4 is published as a short paper in the Empirical Methods in Natural Language Processing (EMNLP) 2018 conference while Chapter 3 is published as a long paper in the Association for Computational Linguistics (ACL) 2019 conference.

1.2 Report Structure

This Thesis’s purpose is to give detailed explorations on the topic of guided abstractive summarization using structural information as well as a new method of manual evaluation that is not referenced-biased. Chapter 1 is the introduction to our work. We then present a literature review of deep learning approaches, automatic summarization, manual and automatic evaluations in summarization, and various approaches on guided summarization in Chapter 2.

Chapter 3 presents our work on a new manual evaluation that is not reference based called Highlight-based Reference-less Evaluation of Summarization (HighRES). We also created a new dataset which is a set of highlighted source documents for the purpose of HighRES evaluation.

Chapter 4 presents our work in guided neural language generation of AMR for summarization. This novel work improved the fluency and informativeness of generated summaries by an AMR summarization system using the source document information.

Chapter 5 presents our work in guided key-phrase extraction for more informative summarization. This work seeks to improve an abstractive summarization system by incorporating key-phrases into the beam-search during the inference process. This work is a negative experiment with mixed-results, however we give insights on why it didn’t work.

Finally, Chapter 6 concludes our work in this Thesis and summarizes our findings for future research.

Chapter 2

Background

In this chapter, we will give the background of automatic summarization, with a specific focus on abstractive summarization, along with evaluation approaches, deep learning approaches for summarization, and how structural information plays a role in the guiding mechanism.

2.1 Deep Learning (DL)

There are many definitions of DL however we chose one definition by Yann Le Cun's¹: “DL is constructing networks of parameterized functional modules & training them from examples using gradient-based optimization. That's it.” This definition fits nicely with many different approaches that we want to discuss later. In this section, we discuss three

¹<https://www.facebook.com/722677142/posts/10156463919392143/>

methods of the deep learning approach that are commonly used for summarizing a text. Some of the following mathematical formulations and figures are adapted from [Goldberg \(2017\)](#)'s book.

2.1.1 Recurrent Neural Networks (RNNs)

RNNs ([Elman, 1990](#)) are feed-forward neural networks where connections between nodes are built upon connected temporal sequences. This characteristic allows RNNs to represent any arbitrary sized sequential input. The following is the formal definition of RNNs. Given an arbitrary length n input sequence of d -dimensional vectors, $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ where $\mathbf{x}_i \in \mathbb{R}^d$, a function called RNN will produce a single hidden vector as an output, \mathbf{y}_n , as denoted by the following equations.

$$\begin{aligned}\mathbf{y}_n &= \text{RNN}(\mathbf{x}_{1:n}) \\ \mathbf{y}_i &= O_{\text{RNN}}(s_i) \\ \mathbf{s}_i &= R_{\text{RNN}}(s_{i-1}, x_i)\end{aligned}\tag{2.1}$$

where O and R are functions that produce \mathbf{y}_i and \mathbf{s}_i as an output and state vectors. The O and R functions for a simple RNN can be defined as identity mapping and activation functions.

Other alternatives that are similar to RNNs are Long Short-Term Memory ([Hochreiter and Schmidhuber, 1997](#), LSTM) and Gated Recurrent Unit ([Cho et al., 2014](#), GRU) networks. These two come under a type of architecture called gated-based architecture and are more commonly used in recent models since they are able to capture long contextual information.

LSTMs can be used for solving the vanishing gradient problem that occurs during a long sequence processing in RNNs. The vanishing gradient problem happens when the value of the gradient keeps diminishing after going through a series of weight multiplications in a long sequence. The LSTMs solution is to consider an additional memory cell besides the state vector s_i . The gating mechanism in LSTM controls how much information is needed to be preserved in memory cells. This allows the gradient to stay high across a long sequence. Formally LSTMs can be defined as follows.

$$\begin{aligned}
\mathbf{s}_i &= R_{\text{LSTM}}(\mathbf{s}_{i-1}, \mathbf{x}_i) = [\mathbf{c}_i; \mathbf{h}_i] \\
\mathbf{y}_i &= O_{\text{LSTM}}(\mathbf{s}_i) = \mathbf{h}_i \\
\mathbf{c}_i &= \mathbf{f} \odot \mathbf{c}_{i-1} + \mathbf{i} \odot \mathbf{z} \\
\mathbf{h}_i &= \mathbf{o} \odot \tanh(\mathbf{c}_j) \\
\mathbf{i} &= \sigma(\mathbf{x}_i \mathbf{W}^{xi} + \mathbf{h}_{i-1} \mathbf{W}^{hi}) \\
\mathbf{f} &= \sigma(\mathbf{x}_i \mathbf{W}^{xf} + \mathbf{h}_{i-1} \mathbf{W}^{hf}) \\
\mathbf{o} &= \sigma(\mathbf{x}_i \mathbf{W}^{xo} + \mathbf{h}_{i-1} \mathbf{W}^{ho}) \\
\mathbf{z} &= \tanh(\mathbf{x}_i \mathbf{W}^{xz} + \mathbf{h}_{i-1} \mathbf{W}^{hz})
\end{aligned} \tag{2.2}$$

where \mathbf{c}_i and \mathbf{h}_i are memory and hidden state component. There are three gates, $\mathbf{i}, \mathbf{f}, \mathbf{o}$ that control how much information can be stored in memory and how much is to be forgotten.

As LSTMs are computationally expensive, GRUs (Cho et al., 2014) is an alternative that is simpler and computationally cheaper (Yang et al., 2020) with comparable performance depending on the type of training dataset. The strength of GRUs comes as GRUs has fewer gates and no separate memory component. The formulation of GRUs is as follows.

$$\begin{aligned}
\mathbf{s}_i &= R_{\text{GRU}}(\mathbf{s}_{i-1}, \mathbf{x}_i) = (1 - \mathbf{z}) \odot \mathbf{s}_{i-1} + \mathbf{z} \odot \tilde{\mathbf{s}}_i \\
\mathbf{y}_i &= O_{\text{GRU}}(\mathbf{s}_i) = \mathbf{h}_i \\
\mathbf{z} &= \sigma(\mathbf{x}_i \mathbf{W}^{xz} + \mathbf{s}_{i-1} \mathbf{W}^{hz}) \\
\mathbf{r} &= \sigma(\mathbf{x}_i \mathbf{W}^{xr} + \mathbf{s}_{i-1} \mathbf{W}^{hr}) \\
\tilde{\mathbf{s}}_i &= \tanh(\mathbf{x}_i \mathbf{W}^{xs} + (\mathbf{R} \odot \mathbf{s}_{i-1} \mathbf{W}^{sg}))
\end{aligned} \tag{2.3}$$

where \mathbf{z} and \mathbf{r} are the gates of GRUs.

In the context of abstractive summarization, one approach that works well is the sequence-to-sequence (seq2seq) (Sutskever et al., 2014) model. A seq2seq model comprises of an encoder and decoder neural network. The seq2seq model can turn a given sequence of tokens into another sequence of tokens of which is very useful for the task of abstractive summarization.

The encoder and decoder network can use RNNs, LSTMs, GRUs or Transformers as the foundation network. The standard seq2seq model's performance however, deteriorates rapidly as the length of the input increases (Cho et al., 2014). This problem can be addressed by using an attention-based mechanism (Bahdanau et al., 2015; Luong et al., 2015). The attention-based mechanism is an alignment model where during the token generation process the model seeks to only attend to a subset of encoded token representations that aligns best with the decoder state at a particular time step. The following is the formulation of a seq2seq with an attention-based mechanism (Luong et al., 2015).

Given an arbitrary length n input of x_1, x_2, \dots, x_n , the encoder network first computes the hidden representation of the input, z_1, z_2, \dots, z_k . Following this, the decoder network generates the target words, y_1, y_2, \dots, y_m , using the conditional probability $P_{s2s}(y_j|y_{<j}, z)$, which is calculated using the equation

$$P_{s2s}(y_j|y_{<j}, z) = \text{softmax}(\mathbf{W}_s \tilde{\mathbf{h}}_t) \quad (2.4)$$

where the attentional hidden state, $\tilde{\mathbf{h}}_t$ is calculated using the equation

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}_c[\mathbf{c}_t; \mathbf{h}_t]) \quad (2.5)$$

where \mathbf{c}_t is the source context vector, and \mathbf{h}_t is the target RNNs hidden state. The source context vector is defined as the weighted average over all source RNNs hidden states, $\bar{\mathbf{h}}_s$, given the alignment vector, \mathbf{a}_t where \mathbf{a}_t is defined as

$$\mathbf{a}_t(s) = \frac{\exp(\text{score}(\mathbf{h}_t, \bar{\mathbf{h}}_s))}{\sum_{s'} \exp(\text{score}(\mathbf{h}_t, \bar{\mathbf{h}}_{s'}))} \quad (2.6)$$

2.1.2 Convolutional Neural Networks (CNNs)

CNNs (LeCun and Bengio, 1995) model work by the principle of convolution and pooling. They are able to capture the representation of local features within a larger structure and combine them into a single hidden vector representing the structure. This architecture was first designed for the Computer Vision field but to this date, it has been successfully

introduced to Natural Language Processing (NLP) field. Here we will explain CNNs within the scope of the NLP field.

The formal definition of CNNs is as follows. Given an arbitrary length n input sequence of d -dimensional vectors with $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ where $\mathbf{x}_i \in \mathbb{R}^d$, we define a function “filter”, as a function that transforms a window of k tokens from input sequence into a single value. The filter function is a linear transformation with weight vector u followed by a nonlinear activation function. It slides across the input sequence resulting in a value \mathbf{p} as follows.

$$\begin{aligned}
 p_i &= g(\mathbf{z} \cdot \mathbf{u}) \\
 \mathbf{z}_i &= \text{concat}(\mathbf{x}_{i:i+k-1})
 \end{aligned}
 \tag{2.7}$$

where g is a nonlinear activation function. The vector p can then be pooled into a single vector c either by max or mean pooling. This vector c represents the whole structure.

Similar to RNN, CNNs also has its own seq2seq version called Convolutional Seq2Seq (Gehring et al., 2017, ConvS2S). In this model, CNNs are used to compute the hidden representation and decoder states. The following will explain the formulation of ConvS2S.

Given an arbitrary length n input sequence of d -dimensional vectors, x_1, x_2, \dots, x_n , we first enrich the input sequence with positional information, p_1, p_2, \dots, p_n where p_i is a positional embedding, before passing them into the encoder. The encoder is a stack of l of one-dimensional convolutional blocks followed by a non-linear activation unit. This stacking mechanism allows ConvS2S to capture a hierarchical representation of the input

sequence where neighbouring words interact in a lower layer while distant words interact in a higher layer. The output of the encoder is the hidden representation of the input sequence, $z_1^l, z_2^l, \dots, z_n^l$. The decoder is a single block with kernel width k . The output of the decoder is the decoder state, $h_1^l, h_2^l, \dots, h_n^l$.

One model that is using ConvS2S as an abstractive summarization model is Topic ConvS2S (Narayan et al., 2018c) which we use for HighRES manual evaluation in Chapter 3.

2.1.3 Transformer Networks

Transformers (Vaswani et al., 2017) work by the principle of stacked self-attention and fully connected layers. The strength of Transformers come from its high parallelization support and long contextual capturing capabilities. Transformer is designed primarily as an encoder-decoder network although it can be used as a stand-alone encoder network as well. We will formulate Transformers as follows.

Consider an arbitrary length n input sequence of d -dimensional vectors with $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ where $\mathbf{x}_i \in \mathbb{R}^d$. The encoder produces hidden representations, z_1, z_2, \dots, z_n from the input sequence and the decoder generates an output sequence, y_1, y_2, \dots, y_m one token at a time. Both encoder and decoder layers comprise of a stack of N identical layers where each stack is defined as follows.

$$\begin{aligned}
\tilde{\mathbf{h}}^l &= \text{LayerNorm}(\mathbf{h}^{l-1} + \text{MultiHeadAttn}(\mathbf{h}^{l-1})) \\
\mathbf{h}^l &= \text{LayerNorm}(\tilde{\mathbf{h}}^l + \text{FFN}(\tilde{\mathbf{h}}^l)) \\
\mathbf{h}^0 &= \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_n\}
\end{aligned} \tag{2.8}$$

where \mathbf{h}^l is the l -th depth layer.

Layer Normalization (Ba et al., 2016), LayerNorm function normalizes layer by computing the mean and variance for all the summed inputs in a layer. It is defined as:

$$\begin{aligned}
\mu &= \frac{1}{l} \sum_{i=1}^l \mathbf{h}^i; b = \frac{1}{d} \sum_{i=1}^d (\mathbf{h}^l - \mu)^2 \\
\text{LayerNorm}(h_l) &= \frac{\mathbf{h}_l - \mu}{\sqrt{b + \epsilon}}
\end{aligned} \tag{2.9}$$

Multi-head Attention represents a number of self-attention in Transformer's architecture. The self-attention mechanism takes its input as three facets: Queries, Keys, and Values. Queries and Keys are of d_k dimensions while Values is of d_v dimensions. These three are then combined as follows.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}V\right) \tag{2.10}$$

The resulting self-attention, called “head”, is then repeated in different positions. The resulting multi-heads are concatenated as follows.

$$\text{MultiHeadAttn}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h)W^O \quad (2.11)$$

where W^O is the output weight for Multi-head Attention.

The decoder has an extra layer that performs multi-head attention over the output of the encoder stack. In addition to that, the self-attention in the decoder only attends to all positions up to the position of the decoder itself to avoid illegal information flow (peeking ahead).

2.2 Pre-trained Language Model

Pre-trained language models (Mikolov et al., 2013; Peters et al., 2018; Radford et al., 2018; Devlin et al., 2019) have been shown to improve performance in many natural language tasks. There are two strategies in using a pre-trained model: feature-based and fine-tuning. A feature-based strategy like word2vec or ELMo (Mikolov et al., 2013; Peters et al., 2018) pre-trains a model on a large unlabeled dataset and then uses it as an additional source of features for the downstream task. Meanwhile fine-tuning (Radford et al., 2018; Devlin et al., 2019) uses the pre-trained model as the basis for further parameter fine-tuning on the downstream task. Since the second strategy has been shown to be the better alternative (Devlin et al., 2019) and also the most commonly used one

in recent research, we focus on approaches that belong to it. In the next subsection we discuss several pre-trained model approaches categorized in how they train the model.

2.2.1 Bidirectional Encoder Representations from Transformers

(BERT)

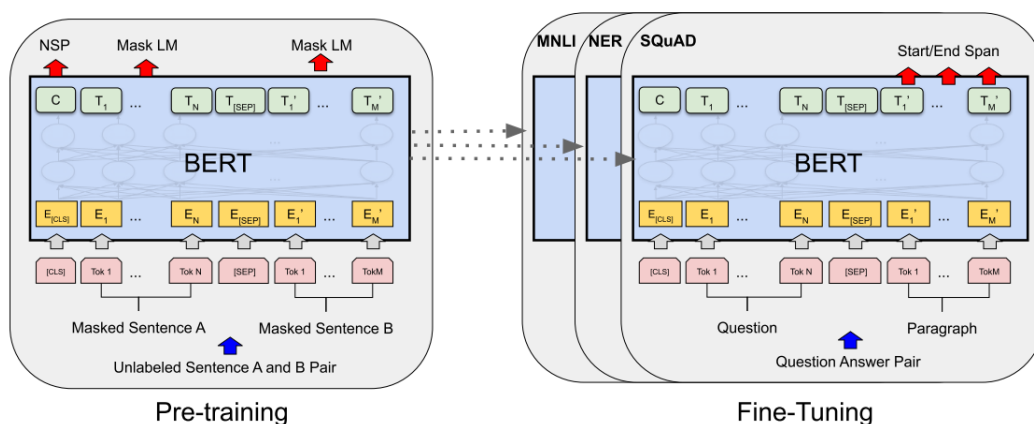


FIGURE 2.1: BERT illustration (Devlin et al., 2019).

There are two steps in BERT: pre-training and fine-tuning steps. In the pre-training step, a large unlabeled dataset is used for training the model. While in the fine-tuning step, the BERT model is initialized with the pre-trained parameters and then fine-tuned on the downstream task. For the input representation, BERT uses WordPiece embeddings (Wu et al., 2016).

For the pre-training step, BERT was trained on two unsupervised tasks: Masked Language Model (MLM) and Next Sentence Prediction (NSP). MLM is done by randomly masking some percentage of the input tokens and then predict those tokens. NSP on the other hand predicts whether the subsequent sentence is the next sentence or a random

sentence. However, BERT is severely under-trained and an alternative RoBERTa (Liu et al., 2019) is proposed to solve it.

RoBERTa proposes improvements over BERT through several modifications as follows.

1. Longer training, bigger batch size and bigger data.
2. Removal of the next sentence prediction objective.
3. Longer input sequence.
4. Mask patterns are dynamically changed when using training data.

The improvements gives RoBERTa a better result over BERT's on both GLUE and SQuAD.

2.2.2 BART

While BERT and RoBERTa are designed to be encoder only pre-trained models, BART (Lewis et al., 2020) is designed as an encoder-decoder pre-trained model and therefore it is more suitable for use in summarization. BART uses the same Transformer architecture with BERT with the exception that it uses GeLU (Hendrycks and Gimpel, 2016) instead of ReLU as the activation layer.

For the pre-training step, BART uses a denoising auto-encoder with several strategies:

1. Token masking: replacing random tokens with masked elements.
2. Token deletion: randomly deleting tokens from the input.
3. Text infilling: randomly masking a span of text. The span length is sampled using a Poisson distribution.
4. Sentence permutation: randomly shuffling sentences.
5. Document rotation: selecting random tokens and rotating them to the beginning of the document.

2.2.3 Text-to-Text Transfer Transformer (T5) approach

The motivation for T5 (Raffel et al., 2019) arises from the need to understand and unify various pre-trained models framework approaches. T5 is similar to BART where it treats the task of pre-training as text-to-text that is the input and output are both in the form of a sequence of tokens.

T5 uses Transformers (Vaswani et al., 2017) as its component with some differences, such as the removal of the Layer Normalization (Ba et al., 2016) bias, layer normalization placement outside the residual path, and using a position embedding scheme of which is a scalar that is added to the corresponding logit used for computing the attention weights. Aside from the difference in the Transformers architecture, T5 also trained on a clean

dataset called The Colossal Clean Crawled Corpus (C4). The C4 text is a clean version of the Common Crawl dataset².

2.2.4 Pegasus

Pegasus (Zhang et al., 2020) follows the same approach as BART (Lewis et al., 2020) and T5 (Raffel et al., 2019) where the input and output of the pre-training task are a sequence of tokens. However, Pegasus uses a new pre-training objective called Gap Sentences Generation (GSG) which is specific to the summarization use. The GSG masked the whole sentence sentences from the document and concatenate the gap sentences into a pseudo summary. There are three approaches that are used to select masked sentences: random, lead and principal. The random approach means sentences are uniformly selected at random, lead approach means only the top- m sentences are selected, and principal approach means sentences that are highly aligned with the rest of the document based on ROUGE scores are selected.

2.3 Automatic Summarization

Formally, automatic summarization is defined as follows. Let $\mathbf{S} = \{s_1, \dots, s_n\}$ denote a sequence of n words of a given input text which is called *source document*, the task is to generate a shorter sequence of words called *target summary*, $\mathbf{Z} = \{z_1, \dots, z_k\}$ where $k < n$. Typically the summary's length is bounded on a given budget, b , which is often

²<https://commoncrawl.org/>

measured by word number or file bytes. Finally, the task’s challenge is to maximize the quality of the produced summary measured by an objective function, f , while limited by a budget. We can further expand this simple definition into different summarization types based on two most common characteristics among many others that are related to our work: the number of source documents and the type of the summary.

2.3.1 Multi-Document and Single Document Summarization

Based on the number of source documents, we can categorize summarization into two different types: *Multi-Document Summarization* (MDS) and *Single Document Summarization* (SDS). MDS (Carbonell and Goldstein, 1998; Fujishige, 2005; Lin and Bilmes, 2012; Kulesza and Taskar, 2016; Perez-Beltrachini and Lapata, 2021) takes a cluster of thematically related source documents, $\mathbf{D} = \{d_1, \dots, d_{|\mathbf{D}|}\}$ as the input and produces a single summary that covers only salient and relevant information from the cluster. This is in contrast with SDS (Rush et al., 2015; See et al., 2017; Lewis et al., 2020; Zhang et al., 2020), where only one document, d , is available as the input. Each type plays an important role in the field and comes with its own set of challenges. We will briefly review challenges and techniques from past works for each type of summarization.

2.3.1.1 Multi-Document Summarization (MDS)

In MDS there are two main objectives (Carbonell and Goldstein, 1998): novelty and relevancy. A good system must be able to produce summaries with high novelty, i.e.

reduced redundancy of information, and relevancy, i.e. covering salient information in accordance to the user's need. Avoiding high redundancy can be done by reducing either partially or fully duplicate information in retrieved summaries. At the same time we want to ensure that retained information has a wide coverage to all salient information in the source document cluster. Furthermore, the produced summaries must also be bounded by a certain length limit. Optimizing these three properties (relevancy, novelty, and length) is an intractable operation as shown in the work of [McDonald \(2009\)](#). Despite the intractability, there are many approaches that can be used to find a near-optimal solution for multi-document summarization such as Submodularity, Integer Linear Programming, Determinantal Point Processes, Sentence Scoring using Regression and others.

2.3.1.2 Single Document Summarization (SDS)

In SDS, there is usually a little repetition of information in the source document, therefore the challenge is more on producing summaries that have the highest informativeness yet are still fluent. Since redundancy issues are not the biggest issue in SDS, many works in SDS determine the informative value by inferring it from regularities of the training dataset. We are going to review several of these approaches as follows.

Neural Network The neural network approach and particularly the one with a sequence-to-sequence architecture is currently the state-of-the-art approach in abstractive summarization. The work is pioneered by [Rush et al. \(2015\)](#) with their Attention-Based Summarization (ABS) and ABS+ systems, which used an adaptation of feed-forward

Neural Network Language Models (NNLM) (Bengio et al., 2003) as the decoder, and as its encoder they used an attention-based contextual model (Bahdanau et al., 2015). For training, Rush et al. (2015) used Stochastic Gradient Descent to minimize the negative log-likelihood. ABS+ system, which is an improved ABS is capable to generate unseen tokens (words that do not exist in the input vocabulary). It also used additional unigram, bigram, and trigram matching features into the standard ABS scoring function. ABS+ system shows a 2.23 ROUGE 1 points improvement over the non neural approach on Gigaword dataset (Graff et al., 2003b).

Chopra et al. (2016) further improved the result of Rush et al. (2015)'s work by 4.02 ROUGE 1 points using their system called Recurrent Attentive Summarizer (RAS) on Gigaword dataset. Chopra et al. (2016) replaced the NNLM core model of Rush et al. (2015)'s system with the encoder-decoder with attention architecture where the encoder is a convolutional networks and the decoder is a recurrent neural networks. Later on, Nallapati et al. (2016) uses a recurrent neural networks encoder-decoder architecture that further improves Chopra et al. (2016)'s result by 1.52 ROUGE 1 points on CNN/DM dataset (Hermann et al., 2015).

See et al. (2017)'s work in automatic summarization further improved Nallapati et al. (2016)'s work by 4.07 ROUGE 1 points on CNN/DM dataset. See et al. (2017) uses a copy mechanism that enables the model to copy from the source document if needed. Subsequent works with the neural model have also shown good results and improved the state-of-the-art several times in a short span of time.

Integer Linear Programming ILP is also used in SDS (Yoshida et al., 2014; Liu et al., 2015). Yoshida et al. (2014) formulated the problem of SDS as a knapsack problem over a dependency-based discourse tree parses of the source document. Similarly, Liu et al. (2015) also formulated the problem as a knapsack problem but over an Abstract Meaning Representation (Banarescu et al., 2013, AMR) graph parses of the source document. In both cases, results are heavily determined by the accuracy of the parser.

2.3.2 Extractive and Abstractive Summarization

Based on the type of the target summary, there are two approaches: *extractive* and *abstractive* summarization. Extractive summarization retrieves a subset of sentences from one or more source documents in their exact form and concatenates them into a summary. Abstractive summarization rephrases source document’s salient information into a more concise summary that may contain new words. Abstractive summarization is considered a more challenging problem compared to the extractive one since it requires an extra step of finding and combining not only textual units but semantic information from the text. Currently, most abstractive summarization systems are done through the use of the seq2seq approach. We will review various techniques of extractive and abstractive summarization in the following subsections.

2.3.2.1 Extractive Summarization

An extractive approach can be posed as a knapsack problem where the task is to select a subset of sentences or sub-sentences from the source document within a limited length budget that collectively have the highest amount of salient information. The challenge is that solving the knapsack formulation is an NP-hard problem, in addition to that determining a correct sentence valuation is also a problem by itself especially for single document summarization where there is very little information redundancy. However, it is difficult to compare these approaches together as each comes with their own evaluation approaches.

We don't work on extractive summarization due to the difficulty of producing a fluent and coherent language using an extractive approach. Moreover, mosts summarization datasets available have abstractive gold summaries in them which makes them a natural fit for an abstractive summarization approach.

The following paragraphs describe how previous works address the issue of extractive summarization.

Sentence Selection Sentence selection ([Carbonell and Goldstein, 1998](#); [Lin and Bilmes, 2011](#); [Ouyang et al., 2011](#); [Cao et al., 2015](#); [Kulesza and Taskar, 2016](#); [Ren et al., 2016](#)) is done by finding a subset within a set of sentences with desirable properties such as feature

scores (Ouyang et al., 2011; Cao et al., 2015; Ren et al., 2016) or low redundancy coverage (Carbonell and Goldstein, 1998; Lin and Bilmes, 2011; Kulesza and Taskar, 2016). However, this approach does not ensure the coherence of the selected sentences.

Sentence Compression Sentence compression (Knight and Marcu, 2002; Almeida and Martins, 2013; Berg-Kirkpatrick et al., 2011) is done by deleting tokens that are not important. Some approaches (Almeida and Martins, 2013; Berg-Kirkpatrick et al., 2011) have two objectives: select sentences and delete words. These models are jointly trained to seek an optimal solution balancing both objectives. Another work like Knight and Marcu (2002) focused instead on single sentence compression. Figure 2.2 shows the illustrations of sentence compression.

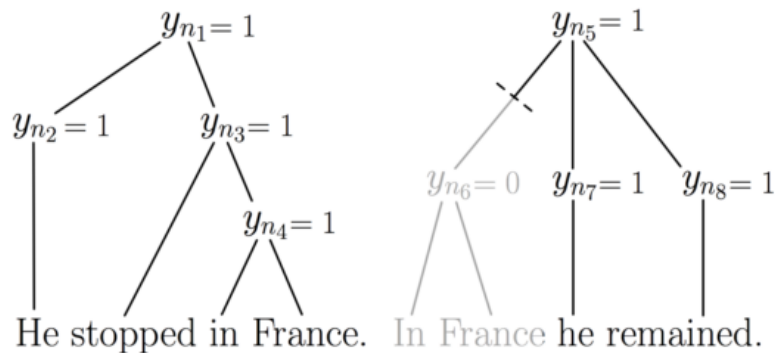


FIGURE 2.2: A compression of two sentences using dependency trees. Dropped words are colored in a lighter color. (Berg-Kirkpatrick et al., 2011).

2.3.2.2 Abstractive Summarization

An abstractive approach produces a summary generated using methods such as paraphrasing, generalization, deductive reasoning, and etc. It is, therefore, more complex

than the extractive approach and requires an extensive understanding of natural language. In recent years, there have been major breakthroughs in the field of abstractive summarization through the use of the seq2seq and pre-trained language models. In the following subsections, we will review several techniques for the abstractive approach.

Sentence Fusion In sentence fusion, the target summary is produced through merging and rephrasing processes of related information. To facilitate the merging process, the approach typically uses intermediary representations such as Dependency Trees (McKeown et al., 1999; Barzilay and McKeown, 2005; Filippova and Strube, 2008) or Abstract Meaning Representation (Liu et al., 2015) among many others. There are however limitations in the sentence fusion approach, i.e.: the intermediate representation parser might not perform well, which introduces noise in the downstream process, and the difficulty to generate fluent sentences from intermediate representations.

Neural model Advances in the seq2seq model with an attention mechanism (Bahdanau et al., 2015; Luong et al., 2015) have seen the surge of automatic summarization approaches. This approach encodes a sequence of tokens from a source document into a hidden representation and then decodes another sequence of tokens as a target summary.

The availability of a large summarization dataset such as CNN/DM (Hermann et al., 2015), Gigaword (Graff et al., 2003a), and XSUM (Narayan et al., 2018c) among many others have helped the progress of neural summarization. The invention of pre-trained language models (Mikolov et al., 2013; Peters et al., 2018; Devlin et al., 2019) and the

seq2seq version of pre-trained language models (Lewis et al., 2020; Zhang et al., 2020) have also improved many state-of-the-art systems. However, the approach still has many limitations such as:

1. Generated summaries are prone to hallucination, grammatical error and other linguistics mistakes (Wiseman et al., 2017).
2. Requires large dataset to perform well.

2.3.3 Summarization using Structural Information

While we showed that the neural approach can achieve state-of-the-art performance in single document summarization, it still has difficulty when dealing with small datasets, for example the DUC and TAC datasets. It also produces summaries that contain hallucinations. As such, many works (Liu et al., 2015; Barzilay and McKeown, 2005) have tried to build summarization systems that are capable to use structural information, and able to work on a small dataset. In the next sections, we will review works in summarization that use structural information.

2.3.3.1 Summarization Using Dependency Trees

Dependency trees were used in previous works such as Barzilay and McKeown (2005); Filippova and Strube (2008); Berg-Kirkpatrick et al. (2011); Almeida and Martins (2013). Barzilay and McKeown (2005); Filippova and Strube (2008) used dependency trees for

sentence fusion while [Berg-Kirkpatrick et al. \(2011\)](#); [Almeida and Martins \(2013\)](#) used them for sentence compression. The use of dependency trees is supported by the availability of a good quality parser such as the Stanford Dependency Parser ([Chen and Manning, 2014](#)). Early works in sentence fusion summarization were done by combining dependency trees using heuristic rules such as in [McKeown et al. \(1999\)](#). These rules were used to discover predicate-argument structures from dependency trees, after which sub-trees that have a similar predicate-argument structure are combined as a resulting summary tree. This approach however is difficult when dealing with sentence paraphrasing since two different dependency trees may share the same meaning but are composed of different words. This limitation has become the major problem in sentences fusion work using dependency trees. Other similar works are [Barzilay and McKeown \(2005\)](#) and [Filippova and Strube \(2008\)](#).

[Barzilay and McKeown \(2005\)](#) performed sentence fusion by synthesizing common information across documents. To identify common information, they developed a method for aligning dependency trees of input sentences. [Barzilay and McKeown \(2005\)](#)'s work comprised of three steps:

1. Identification of common information is done by aligning the similarity between the structure of the dependency trees and the similarity between lexical items of the input sentences.
2. Fusion lattice computation which combines intersection subtrees.

3. Lattice linearization into text which includes a selection of a tree traversal order, lexical choice among available alternatives, and placement of auxiliaries, such as determiners.

[Filippova and Strube \(2008\)](#) uses compression where they combined all the input trees as a single tree then compress the union tree. In this way, all information from the sentence will be included in the union tree instead of relying on a single basis tree as in [Barzilay and McKeown \(2005\)](#)'s work.

Other sentence compression approaches are [Clarke and Lapata \(2008a\)](#), [Berg-Kirkpatrick et al. \(2011\)](#) and [Almeida and Martins \(2013\)](#). They used Integer Linear Programming to formulate the compression problem where dependency trees are used to ensure the result is grammatically sound. This can be done by ensuring the head word and modifier relation remain grammatical after compression. The limitation of this approach is that it can't bring together two textual units (phrases or words) that are located far apart within the document but are still related to each other. This issue is a major challenge when one wants to build an abstractive summarization system.

2.3.3.2 Summarization Using Abstract Meaning Representation

An AMR graph ([Banarescu et al., 2013](#)) is a rooted, directed, and labelled graph that captures the semantic representation of '*who is doing what to whom*' in a sentence (see [Figure 2.3](#)). It can also be represented in PENMAN format([Matthiessen and Bateman, 1991](#)) as follows.

```
(w / want-01
  :ARG0 (b / boy)
  :ARG1 (b2 / believe-01
    :ARG0 (g / girl)
    :ARG1 b))
```

One advantage of using AMR is that it abstracts away from syntactic idiosyncrasies for example, sentences like “*The man described the mission is a disaster.*” and “*As the man described it, the mission is a disaster.*” will be represented by the same following AMR:

```
(d / describe-01)
  :arg0 (m / man)
  :arg1 (m2 / mission)
  :arg2 (d / disaster))
```

Besides AMR there are other meaning representation formats such as Discourse Representation Structures ([Abzianidze et al., 2020](#)), Semantic Dependency Parsing ([Oepen et al., 2014](#)), FrameNet [Fillmore and Baker \(2010\)](#), and many others, however, only AMR provides a gold label summarization dataset which is why we use AMR in our work.

When AMR graphs are used in summarization ([Liu et al., 2015](#)) to represent the semantic interaction between words, we can focus on capturing semantic information and process them while staying agnostic to the syntactic structure of the input, for example in the case of sentence paraphrasing.



FIGURE 2.3: An AMR Graph of the sentence ‘the boy wants the girl to believe him.’ (Liu et al., 2015) There is a re-entrancy in the concept node of ‘boy’ which has several parents.

Liu et al. (2015) introduced AMR graph manipulation for summarizing text as a series of processes such as graph merging and sub-graph prediction. While graph merging is a heuristic approach, sub-graph prediction can be formulated as an ILP process where parameters can be learned in a supervised manner. Liu et al. (2015) however didn’t perform AMR-to-text generation, instead, they opted to generate bag-of-words that are sorted based on each word occurrence in the source document. To evaluate their results, the subgraph of the prediction summaries are compared against the gold summary graphs. This is possible as the AMR dataset (Knight et al., 2017) provides gold labels AMR summary graphs.

Even though the results are not fluent, they showed that their approach achieves 58.7% accuracy in node matching between gold summary AMR parses and system AMR predictions. In the next section, we show how graph merging and sub-graph prediction are done in Liu et al. (2015).

In graph merging, parses of input sentences AMR graph were combined into a single graph called source graph (see Figure 2.4), $G = (V, E)$ where $v \in V$ and $e \in E$ are unique

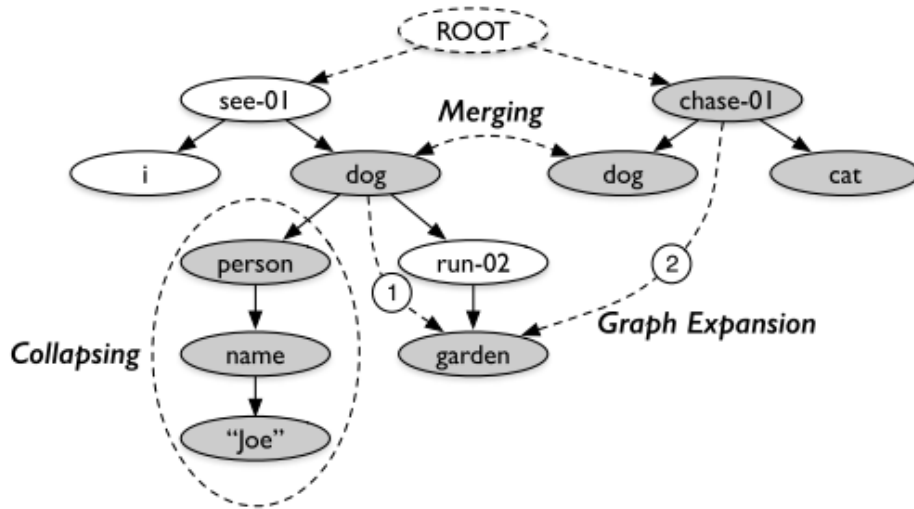


FIGURE 2.4: A joined AMR graph from two sentences AMR graphs. (1) and (2) are added edges through expansion steps. (Liu et al., 2015)

concepts and relations between pairs of concepts. The merging process is a heuristic process where a pair of nodes that belongs to the same label from two different AMR graphs are combined into one node. Afterwards, finding a smaller graph called summary graph, G' , can be formulated as an ILP problem as follows.

$$\begin{aligned}
& \text{maximize } \sum_{i=1}^N v_i \boldsymbol{\theta}^\top \mathbf{f}(i) + \sum_{(i,j) \in E} e_{i,j} \boldsymbol{\psi}^\top \mathbf{g}(i,j) \\
& \text{s.t.} \\
& (1) \ v_i - e_{i,j} \geq 0, v_j - e_{i,j} \geq 0, \forall i, j \leq N \\
& (2) \ \sum_i f_{0,i} - \sum_i v_i = 0 \\
& (3) \ \sum_i f_{i,j} - \sum_k f_{j,k} - v_j = 0, \forall j \leq N \\
& (4) \ N \cdot e_{i,j} - f_{i,j} \geq 0, \forall i, j \leq N \\
& (5) \ \sum_j e_{i,j} \leq 1, \forall j \leq N \\
& (6) \ \sum_i \sum_j e_{i,j} = L
\end{aligned} \tag{2.12}$$

where $\mathbf{f}(i)$ and $\mathbf{g}(i, j)$ are feature representations of node i and edge (i, j) respectively, $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ are parameters that have to be learned from the training data, v_i and $e_{i,j}$ are selection indicators. Constraint (1) is to ensure the validity of the graph, constraints (2) - (4) is to ensure the graph connectivity, constraint (5) is to ensure that the resulting graph is a tree, and finally constraint (6) bounds the graph size.

2.3.4 Summarization Using a Guiding Mechanism

In this section, we discuss approaches where structural information (See et al., 2017; Li et al., 2018a; Jin et al., 2020; Li et al., 2020; Dou et al., 2020) can be used as a guidance

signal. A guidance signal can be defined as different types of signal that are fed into the model in addition to the source document (Dou et al., 2020). In the following subsections, we will discuss several works that use the guiding mechanism.

Li et al. (2018a)’s System While See et al. (2017)’s approach is very good in handling the OOV problem, however it is hard for Pointer Network to correctly identify which words are to be copied. Li et al. (2018a) addressed this problem by proposing a system called Key Information Guide Network (KIGN). KIGN is a system where the model would accept external knowledge in the form of key-phrases of which are extracted prior to the training using TextRank (Mihalcea and Tarau, 2004). These key-phrases become part of the model input as external information and are encoded using a separate bi-LSTM layer. The encoded key-phrases are then used as part of the model attention and also in the pointer mechanism. The modified pointer mechanism is as follows.

$$p_{\text{gen}} = \sigma(h_t^*, s_t, k_t) \quad (2.13)$$

where k_t is the encoded key-phrase in time t . The KIGN model with guidance shows an improvement of 2.51 ROUGE 1 points higher than Pointer Generator network (See et al., 2017) on CNN/DM dataset (Hermann et al., 2015).

An example of Li et al. (2018b)’s output can be seen in Table 2.1.

Text(truncated): google claims to have cracked a problem that has flummoxed anyone who has tried to read a doctor’s note - how to read anyone’s handwriting. the firm claims the latest update to its android handsets can under 82 languages in 20 distinct scripts, and works with both printed and cursive writing input with or without a stylus. it even allows users to simply draw emoji they want to send. scroll down for video. the california search giant claims the latest update to its android handsets can understand handwriting in 82 languages in 20 distinct scripts. google says its handwriting recognition works by building on large-scale language modeling, robust multi-language ocr.
Gold: google handwriting input works on android phones and tablets. handsets can under 82 languages in 20 distinct scripts. works with both printed and cursive writing input with or without a stylus
Baseline+pointer-mechanism: google claims to have cracked a problem that has flummoxed anyone who has tried to read a doctor ’s note how to read anyone ’s handwriting
Li et al. (2018b) model: google claims the latest update to its android handsets can under 82 languages in 20 distinct scripts, and works with both printed and cursive writing input with or without a stylus.

TABLE 2.1: Comparison of Li et al. (2018b) model’s output, the pointer generator baseline and the gold summary.

Dou et al. (2020)’s system The guidance system of Dou et al. (2020)’s proposes several types of signals:

1. Highlighted sentences in the source document which are extracted sentences using the greedy approach to maximize the ROUGE score.
2. Retrieved summaries which are summaries of other similar documents extracted using Elastic Search ³.
3. Keywords are a set of salient individual words from the source document extracted using TextRank (Mihalcea and Tarau, 2004)

³<https://github.com/elastic/elasticsearch>

4. Salient relational triples which are triples containing subject, relation and object which are extracted using Stanford OpenIE ⁴.

Guidance signals are then encoded separately using Transformers in the same way as the encoding of the input sequence. Guidance encoded signals are then incorporated into the decoder’s attention mechanism to guide the output. This can be seen in the following equations.

$$\begin{aligned}
 \tilde{\mathbf{h}}^l &= \text{LayerNorm}(\mathbf{h}^{l-1} + \text{MultiHeadCrossAttn}(\mathbf{h}^{l-1}), \mathbf{x}) \\
 \tilde{\mathbf{h}}^l &= \text{LayerNorm}(\mathbf{h}^{l-1} + \text{MultiHeadCrossAttn}(\mathbf{h}^{l-1}), \mathbf{g}) \\
 \mathbf{h}^l &= \text{LayerNorm}(\tilde{\mathbf{h}}^l + \text{FFN}(\tilde{\mathbf{h}}^l)) \\
 \mathbf{h}^0 &= \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_n\}
 \end{aligned} \tag{2.14}$$

where \mathbf{x} and \mathbf{g} are encoded input and encoded guidance signals. However, the Dou et al. (2020)’s model results is only comparable with the baseline BART model (Lewis et al., 2020). Dou et al. (2020)’s model with different guidance signals can be seen in Figure 2.5

⁴<https://nlp.stanford.edu/software/openie.html>

Model	Guidance	Output
Ref.	-	nathan dailo has found a way to get his son to sleep in 42 seconds. in a youtube video he demonstrates how stroking his 3-month-old son's face with a white piece of tissue paper sends him to sleep. the video has received almost 26,000 views in just two weeks.
Sentence	nathan dailo, from sydney, uploaded a video to his youtube channel demonstrating how he gets his three-month-old son seth to drift off in just 42 seconds. the clip that has now received almost 26,000 views sees the father gliding a piece of white tissue paper over his son's face repeatedly until he nods off. in his youtube video, nathan dailo shows that by running tissue paper over his son seth makes him sleep.	nathan dailo, from sydney, uploaded a video to his youtube channel demonstrating how he gets his three-month-old son seth to sleep in just 42 seconds. the clip has now received almost 26,000 views. by running tissue paper over his son seth makes him sleep.
	the clip that has now received almost 26,000 views sees the father gliding a piece of white tissue paper over his son's face repeatedly until he nods off. in his youtube video, nathan dailo shows that by running tissue paper over his son seth makes him sleep. one grateful viewer of the video commented: 'this worked for my baby 5 times already so thanks!'	nathan dailo, from sydney, shows that by running tissue paper over his son seth makes him sleep in 42 seconds. the clip has now received almost 26,000 views. one viewer commented: 'this worked for my baby 5 times already so thanks!'
Keyword	video sleep dailo youtube way seconds demonstrates weeks piece received	nathan dailo, from sydney, uploaded a video to his youtube channel. he demonstrates how he gets his three-month-old son seth to sleep in just 42 seconds. the clip has now received almost 26,000 views since being uploaded two weeks ago.
	son paper sends found demonstrates way views face dailo piece	nathan dailo, from sydney, posted a video to his youtube channel. he demonstrates how he gets his three-month-old son seth to sleep in just 42 seconds. the father glides a piece of white tissue paper over his son's face repeatedly until he nods off.
Relation	(father, gliding, piece of tissue paper), (clip, sees father over, his son's face)	a father from sydney has worked out a way of getting his three-month-old son to sleep. in his youtube video, nathan dailo shows that by running tissue paper over his son seth makes him sleep in just 42 seconds. the clip has now received almost 26,000 views since being uploaded two weeks ago.
	(nathan dailo, uploaded, video to his youtube channel), (has drifted, has received, 26,000 views)	nathan dailo, from sydney, uploaded a video to his youtube channel. the clip has received 26,000 views since being uploaded two weeks ago. it shows the father gliding a piece of white tissue paper over his son's face repeatedly until he nodded off.
Retrieve	phil smith posted the video of him. his son noah's tooth to his youtube channel which went viral with over 100,000 views. smith tied noah's tooth to dental attached to a golf ball. smith hits the ball and the tooth is gone in just seconds, made by the expression in noah's face.	nathan dailo, from sydney, posted a video to his youtube channel. he demonstrates how he gets his three-month-old son to sleep in just 42 seconds. the clip has now received 26,000 views since being uploaded two weeks ago.
	karen wiltshire, 46, sends the babies to sleep by their and playing them the sound of a. it can take anything from a few minutes to half an hour for the tiny newborns to off to sleep. mother-of-two from poole, dorset, has won an award in studio children photography for her work.	mother-of-two from sydney, has won an award in studio children photography for her work. in a youtube video, nathan dailo shows how he gets his three-month-old son seth to sleep in just 42 seconds. the clip has now received 26,000 views since being uploaded two weeks ago.

FIGURE 2.5: Different guidance signals and its corresponding output using [Dou et al. \(2020\)](#)'s model

2.4 Guidance by Constraining Beam Search

Previous approaches in guiding abstractive summarization systems require training the summarization model to include the guidance signal. If during inference time, the domain of the guidance signal changes then a model re-training is necessary. In this section, we will discuss various approaches that don't require model re-training.

2.4.1 Usage in Neural Machine Translation (NMT)

We will first discuss approaches in the NMT field since these approaches are heavily used in that field for correcting mistakes or forcing certain translations.

Chatterjee et al. (2017)’s System The challenge of adding external knowledge in NMT is that words and sentences are represented as continuous representations, which makes it hard to specifically determine how and when one should include external knowledge. To address the issue, Chatterjee et al. (2017) explored the possibility of including external knowledge to the NMT during the decoding process. Their work improved the existing baseline at least 3 BLEU points.

In detail, Chatterjee et al. (2017) used a translation recommendation that comes with the source document to guide the decoder, where using it one can ensure the presence of certain words from the translation recommendation in correct positions of a target sentence. This can be done in three ways, which are reviewed in detail as follows.

Forcing the presence of a given term Forcing can be done by modifying the beam search in the decoding process. Each most probable target word in the probability distribution is checked whether there is a suggested word that is provided through a phrase table that can replace said word. If it is found, the word from the phrase table is used instead of the most probable one.

Placing the term in the right position This is done when suggestion words from the translation recommendation are known to be close to each other and we want to ensure their co-occurrence in the target sentence. An example is when translating “*application*” and the suggestion phrase is “*die Anwendung*”. By using a look-ahead process, a part of the suggested word is checked in the beam decoding process to decide whether “*Anwendung*” could be generated in the near k -steps in the future. If it is not reachable, then a forced replacement is done to force the phrase in the target sentence.

Guiding the Out Of Vocabulary (OOV) terms To handle the issue of OOV terms, a lookup table that stores all the OOV suggestions is used instead of a standard “UNK” token.

Zhang et al. (2018)’s System In Zhang et al. (2018)’s work, they also explored modifications to the decoder of a seq2seq model similar to Chatterjee et al. (2017) however with a different kind of external knowledge. Instead of using a preset translation recommendation, they used a search engine to retrieve sentences and their translation (referred to as translation pieces) that have a high similarity score with the source sentence. When similar n-grams from a source document were found in the translation pieces, they rewarded the presence of those n-grams during the decoding process through a scoring mechanism that calculates the similarity score between the source sentence and source side of the translation pieces. Zhang et al. (2018) reported improvements in translation results up to 6 BLEU points over their seq2seq NMT baseline. In this Thesis, we use

the same principle and reward n-grams that are found in the source document during the AMR-to-Text generation process.

He et al. (2021) In [He et al. \(2021\)](#)'s work, a translation memory which is a source-target pair that is the most similar to the source-target pair to be translated, is encoded using Transformers network and combined with the translation model which is also using Transformers network. There are three approaches of translation memory encoding that are used by [He et al. \(2021\)](#). The first one is embedding the translation memory sequence directly. The second one is to use a scaled embedding where the scaling is done based on the similarity score between the translation memory and the input source sequence. The last one is to use word alignment between the translation memory source and target sequence. [He et al. \(2021\)](#)'s model shows an improvement of 4.47 BLEU points over the baseline Transformers.

2.4.2 Lexically Constrained Decoding

In the previous sections, we discussed several approaches in NMT where certain keywords could be forced to occur in the generated hypothesis by looking at the model attentions of the source document. Other approaches ([Hokamp and Liu, 2017](#); [Post and Vilar, 2018](#)) integrated the keyword look-up in the beam-search by replacing the top- k sampling mechanism of a standard beam-search.

Hokamp and Liu (2017)’s System Hokamp and Liu (2017) introduced the Grid Beam Search (GBS) algorithm which allows the inclusion of pre-specified lexical constraints into the output of the generation process. Although GBS was intended for NMT usage, the principle can be applied to any beam-search process.

The idea of GBS is to allocate $C + 1$ separate beams (banks) that track which constraints are satisfied. There are two variables, t and c , that index and track the beam in the grid. The t variable tracks the time-step of the search while the c variable tracks the constraints that are satisfied in the current beam. In each time step, the beam will update the constraints and build new hypotheses that could satisfy all constraints. After the beam-search process has finished, the beam with the highest number of satisfied constraints is returned. The limitation of this approach is that it requires a large number of additional beams especially when the number of constraints is large. Another issue is that GBS changes the number of beams in each sentence as each sentence has a different number of constraints. As such the work is difficult to be optimized as a batching process in the GPU.

Post and Vilar (2018)’s System Post and Vilar (2018) improved GBS’s limitation by introducing the Dynamic Beam Allocation (DBA) mechanism which works the same way as GBS but uses a fixed k size beam throughout the whole process. DBA is an allocation strategy for distributing hypotheses that satisfy the constraints. The strategy is to reserve the same number of hypotheses that satisfy the same number of constraints in the same bank.

2.5 Automatic and Manual Evaluation in Summarization

The earliest effort ([Edmundson, 1969](#)) to evaluate automatic summarization was done by having human annotators manually judge the quality of the system produced summary in terms of how similar it is with the gold summary, and how informative is the information contained in the summary. At that time, there wasn't any consensus in the summarization evaluation. It was in the DUC 2001 that the evaluation of summarization became the focus of researchers and was continuously getting refined for the subsequent years of the conference. This section is dedicated to giving an overview of automatic and manual summarization evaluation that results from the continuous refinement of summarization evaluation in the DUC and TAC conferences. But first, we are going to review the criteria that are commonly used in summarization.

2.5.1 Common Criteria for Summarization

In determining criteria for summarization, we have to first determine the purpose of evaluating a summary. At a high level, there are two purposes in evaluating a natural processing system ([Jones and Galliers, 1995](#)):

1. evaluating the objective of the system, and
2. evaluating the function of the system relating to an external task.

The first one is called an intrinsic evaluation wherein summarization's case, the objective that needs to be evaluated is the summary intrinsic quality such as fluency, coherence, informativeness and etc., while the second one is called an extrinsic evaluation of which it measures how well the summarization system works in a certain task, for example: assuming that a summary contains all salient information from the document, we can use a Q&A approach to measure how good the summary is by its capability to answer questions that are derived from the source document as shown by [Gorinski and Lapata \(2015\)](#)'s work. As extrinsic evaluation requires another set of external task to evaluate the summary of which create more complexity to the existing evaluation task, we will only review criteria relating to intrinsic evaluation which are more related to our task.

Coherence and Readability In an extractive summarization, multiples sentences are extracted from the source document as a summary. This, however, poses a problem when each sentence is extracted without regarding the context of the other extracted sentences, hence such a summary is likely to have a low coherence. Most works such as [Paulus et al. \(2018\)](#) only use readability where the coherence criterion is implied in it. Readability is a high-level criterion where judges have to rate many aspects at once such as sentence coherence, fluency and naturalness, the presence of dangling anaphors and etc.

Informativeness A summary that covers a lot of information from the source document is deemed informative. Of course, the challenge here is to keep the length of the summary as short as possible while making sure that it has enough coverage. The informativeness

criterion is also called coverage, or content quality. ROUGE (Lin, 2004) and Pyramid (Nenkova and Passonneau, 2004) automatic metrics measure informativeness based on information recall between the reference and the system summary.

Correctness A summary needs to ensure the correctness of the information itself. This is a problem for advances in summarization that used deep learning (See et al., 2017; Paulus et al., 2018) where often the correctness of the information might be lost during the generation process, for example wrong mentions of numbers, dates, and facts. Other studies such as Maynez et al. (2020) and Kryscinski et al. (2020) explore correctness evaluation and propose correctness evaluation methods based on textual entailment (Maynez et al., 2020) and weakly-supervised (Kryscinski et al., 2020) approaches.

Compression Works (Clarke and Lapata, 2008b; Berg-Kirkpatrick et al., 2011; Almeida and Martins, 2013) that focus on summarization through compression also used compression as the criterion to measure the quality of the summary where it is commonly measured as the ratio of the compressed sentence with respect to the original sentence.

2.5.2 Automatic Evaluation

A manual evaluation needs human judges and the process can be long and expensive. Due to this reason, automatic evaluation has become the tool for providing insights into system performance before opting to use manual evaluation. There are many automatic evaluation methods for evaluating summarization results. We will review two of the most

common ones: Recall-Oriented Understudy for Gisting Evaluation (Lin, 2004, ROUGE) which is the most popular one and the Pyramid (Nenkova and Passonneau, 2004) method.

ROUGE ROUGE was specifically designed for evaluating automatic summarization without involving human judgments. Commonly used versions of ROUGE metrics are ROUGE-N and ROUGE-L. The following will describe each of those in detail.

ROUGE-N evaluates a candidate summary based on the n-gram recall between the candidate summary and a set of reference summaries. The formula to calculate ROUGE-N can be seen in Equation 2.15

$$\text{ROUGE-N} = \frac{\sum_{S \in \{\text{Reference Summaries}\}} \sum_{\text{gram}_n \in S} \text{Count}_{\text{match}}(\text{gram}_n)}{\sum_{S \in \{\text{Reference Summaries}\}} \sum_{\text{gram}_n \in S} \text{Count}(\text{gram}_n)} \quad (2.15)$$

Where n stands for n-gram length, gram_n and $\text{Count}_{\text{match}}(\text{gram}_n)$ is the maximum number of n-grams co-occurring in a candidate summary and a set of reference summaries.

ROUGE-L evaluates a candidate summary based on the longest common subsequence between the candidate summary and a set of reference summaries. The formula to calculate ROUGE-L can be seen in Equation 2.16, 2.17 and 2.18.

$$R_{lcs} = \frac{LCS(X, Y)}{m} \quad (2.16)$$

$$P_{lcs} = \frac{LCS(X, Y)}{n} \quad (2.17)$$

$$F_{lcs} = \frac{(1 + \beta^2)R_{lcs}P_{lcs}}{R_{lcs} + \beta^2P_{lcs}} \quad (2.18)$$

Where X of length m is a reference summary, and Y of length n is a candidate summary, $LCS(X, Y)$ denotes the length of a longest common subsequence of X and Y , $\beta = P_{lcs}/R_{lcs}$ when $\partial F_{lcs}/\partial R_{lcs} = \partial F_{lcs}/\partial P_{lcs}$.

Pyramid Pyramid (Nenkova and Passonneau, 2004) is based on specific textual units that are not bigger than a clause called Summary Content Units (SCU). The following examples are given by Nenkova and Passonneau (2004) to show SCU as parts of a sentence.

A1 In 1998 two Libyans indicted in 1991 for the Lockerbie bombing were still in Libya.

B1 Two Libyans were indicted in 1991 for blowing up a Pan Am jumbo jet over Lockerbie, Scotland in 1988.

C1 Two Libyans, accused by the United States and Britain of bombing a New York bound Pan Am jet over Lockerbie, Scotland in 1988, killing 270 people, for 10 years were harbored by Libya who claimed the suspects could not get a fair trial in America or Britain.

D2 Two Libyan suspects were indicted in 1991.

For most of these cases, SCUs in the above examples are spans selected by a human. If there is a sentence segment that occurs in more than two different sentences, it will be

annotated as an SCU. Each SCU is weighted based on the number of occurrences in the examples. The optimal content score for a summary with X SCUs is calculated by the following equation:

$$Max = \sum_{i=j+1}^n \times |T_i| + j \times (X - \sum_{i=j+1}^n |T_i|) \quad (2.19)$$

where $j = \max_i \left(\sum_{t=1}^n |T - i| \geq X \right)$

2.5.3 Manual Evaluation

An automatic evaluation such as Pyramid or ROUGE is lacking in several ways Pyramid still needs human annotation to provide the SCU annotation beforehand and ROUGE is prone to fluency and correctness issues where a bad summary can get a good ROUGE score and the other way around: good summaries can get bad ROUGE score. As such many works still rely on manual evaluation for their summarization system. However, there is still no agreement on how to perform a manual evaluation on summarization, and as such, we will review many different types of manual evaluation in this section.

2.5.3.1 The DUC 2001 Manual Evaluation

The DUC 2001 provides three tasks for participants: single document, multi-document, and exploratory summarization. The single and multi-document summarization were the core tasks of the conference and received 11 and 12 systems submissions respectively. The following paragraphs describe how the manual evaluation was done ([Lin and Hovy, 2002](#)).

Evaluation Materials DUC 2001 provided 10 document clusters making up to 60 set of documents. For each document and the document cluster, there were three human summaries created as references where one of them is the main reference and another two are extra references which were also used for the evaluation purpose. In addition to that, there were the lead-baseline and coverage baseline summaries provided. The lead-baseline summary took the first 50, 100, 200 and 400 words from the document (single) or the last document in the collection (multi-document). The coverage baseline, which was only available in the multi-document dataset, comes from the beginning lines of each document in the document cluster until it reaches a summary of 50, 100, 200 or 400 words.

The two extra references have different content compared to the main reference. The average unigram overlap between the two extra references and the main reference in 50-words summaries is 15.1%, while for 200-words summary is 19.7%, which are very low. But [Harman and Over \(2004\)](#) pointed out that this low number is due to the subjectivity of the human annotators in terms of summary details and granularity.

Evaluation Environment and Metrics To evaluate manually, each human judge used the Summary Evaluation Environment (SEE) 2.0 ([Lin, 2001](#)) which is shown in [Figure 2.6](#). Using SEE, each judge can compare a pair of summaries consisting of reference (model) and system (peer) summaries based on their content and quality. For evaluating content (informativeness), a judge would mark sentences from a reference and system summaries that share similar information and then specify the rating on four weights: 1 for *all*, 0.75 for *most*, 0.5 for *some*, 0.25 for *hardly any*. Once all pairs of similar sentences

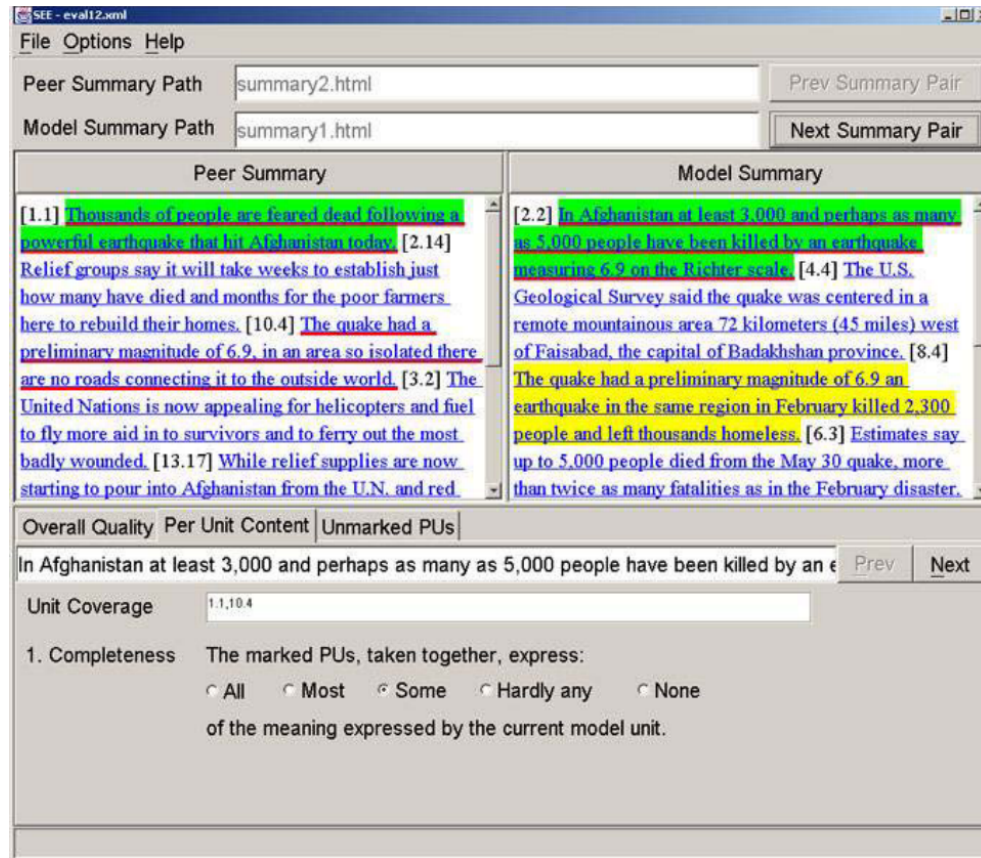


FIGURE 2.6: The Summary Evaluation Environment 2.0 user interface. (Lin, 2001)

were marked and rated, the weighted retention can be measured as:

$$\text{Retention} = \frac{\text{the number of marked sentences}}{\text{the number of marked sentences in all summary examples}} \quad (2.20)$$

For evaluating summary quality, three questions were asked: “Does the summary observe English grammatical rules independent of its content?”, “Do sentences in the summary fit in with their surrounding sentences?”, and “Is the content of the summary expressed and organized in an effective way?” The three questions measure grammatical-ity, cohesion and coherence respectively, with each rated on a five-point scale: *all*, *most*, *some*, *hardly any*, *none*

In [Lin and Hovy \(2002\)](#)'s analysis, it was shown that judges are sometimes inconsistent in assigning scores, where in single document summarization 18% of total judgments are ambiguous (multiple scoring for the same sentence), while in multi-document summarization 7.6% of total judgments are ambiguous. Even though the number of inconsistencies is small, it is still deemed as problematic since in extractive summarization, every similar pair should have the same score as nothing is changed in the sentence structure. Despite the ambiguity, this did not affect the ranking since human summaries were still considered better than summaries produced by the system. However, the ranking might change between systems that have a close scoring when inconsistencies of the scoring are considered.

The manual evaluation result was also compared against an automatic evaluation result for each system. The automatic evaluation is evaluated using n-gram interpolation. The n-gram interpolation is formulated as follows.

$$a_1 * \text{NAM}_1 + a_2 * \text{NAM}_2 + a_3 * \text{NAM}_3 + a_4 * \text{NAM}_4 \quad (2.21)$$

where NAM_n is the n-gram percentage overlapping between system and all references, and a_1 to a_4 are hyper-parameters. The Spearman rank-order correlation between the manual and automatic summary was over 97%. The high correlation result is because of the extractive summarization approaches considered where paraphrasing is not involved.

2.5.3.2 The DUC 2002 Manual Evaluation

The DUC 2002 is the continuation of the DUC 2001 where more sets were added to the dataset and had several improvements over the manual evaluation method.

Evaluation Materials The number of document sets were now twice the amount of the DUC 2001 dataset. The summaries were limited to 200 words, eliminating the previous 400-word summaries.

Evaluation Environment and Metrics The SEE GUI had been updated to replace the text label for the intervals with straight percentages: 20, 40, 60, 80, 100. This was done to help the perception of the judges of the coverage completeness as words might be perceived differently by different judges. In total, there were 13 submitted systems for the single summarization track and 8 submitted systems for the multi-document summarization track.

The manual environment mechanism changed in the way judges evaluate system summaries. Previously, a judge couldn't access all system summaries, which created inconsistencies. This was addressed in the DUC 2002 by giving a judge access to all summaries of a given document, this way all judgments can be given in a consistent manner. The judge can still be more lenient or stricter for the whole document set, of which the DUC 2002 assigned more judges to allow averaging over the effect of strict/lenient judges.

2.5.3.3 The DUC 2003 and DUC 2004 Manual Evaluation

The DUC 2003 and 2004 introduced new tasks that are different compared to DUC 2001 and 2002. The manual evaluation approach was also changed which comes from experience running previous years' DUC.

Evaluation Materials DUC 2003 introduced four tasks for summarization: one very short summary for a single document, which is intended to generate headlines in newswire, and three different multi-document summarization tasks. DUC 2004 introduced two additional tasks of summarization in Arabic and the fifth task which is a blend of task 3 and 4 of DUC 2003.

Evaluation Environment and Metrics The SEE GUI had been updated again for this DUC. In DUC 2003, there are now 12 questions prepared for measuring the summary quality (see Appendix A) which can be answered by choosing one of four options: 0, 1-5, 6-10, >10 while in DUC 2004, more complex questions are presented to the judges (see Appendix B).

The coverage questions were also changed with only one question (previous years are 3 questions) using the wording: *“When you have marked all such PUs for the current MU, then think about the whole set of marked PUs and answer the question:”, “The marked PUs, taken together, express about [0%, 20%, 40%, 60%, 80% and 100%] of the meaning expressed by the current reference unit.”*

Besides the summary quality and the coverage quality from the previous DUCs, judges also evaluated the usefulness of the summary in this year DUC. In this evaluation, judges were given a document and all summaries of that document. Judges were then asked this question: *“Assume the document is one you should read. Grade each summary according to how useful you think in getting you to choose the document.”*. judges would then rank each summary from 0 (worst, of no use) to 4 (best).

2.5.3.4 The DUC 2005-2007 and TAC 2008-2009 Manual Evaluation

Starting from the DUC 2005 and afterwards, both tasks and manual evaluation for the summarization track were simplified. The DUC 2005 itself is the starting point of finding new evaluation methods that take into account variations in the gold summaries' content. The next DUC eventually changed into Text Analysis Conference (TAC) which subsequently improves from the DUC 2005 framework.

Evaluation Materials The DUC 2005 and 2006 had only one task that is query-focused summarization. DUC 2007 had two generic summarization tasks which are the multi-document 250-word (TAC 2008 and afterwards reduce this to 100-word) summary as the main task and the 100-word summary as the update task. The update task is intended to challenge a system to produce a summary after the model has read the main task. The TAC 2008 and 2009 had an update task and an opinion summarization task.

Evaluation Environment and Metric Starting in the DUC 2005, the metrics for manual evaluations were changed into two metrics: readability and responsiveness. The readability was evaluated based on five linguistic properties: grammatical, non-redundancy, referential clarity, focus, and structure + coherence. See Appendix C for questions regarding the readability metrics.

2.5.3.5 Post DUC and TAC

After DUC and TAC, summarization literature has investigated different means of conducting the manual evaluation. We studied a sample of papers from major ACL conferences and outline the trends of manual evaluation in summarization in Table 2.2. A majority of work has focused on evaluating the content and the linguistic quality of summaries. However, there seems to be a lack of consensus on how a summary should be evaluated: (i) Should it be evaluated relative to other summaries or standalone in absolute terms? and (ii) What would be a good source of comparison: the input document or the reference summary? The disagreements on these issues result in authors evaluating their summaries often (11 out of 26 papers) using automatic measures such as ROUGE (Lin, 2004) despite of its limitations (Schluter, 2017). In what follows, we discuss previously proposed approaches along three axes: evaluation metrics, relative vs. absolute, and the choice of reference.

Evaluation Metrics Despite differences in the exact definitions, the majority (e.g., Hsu et al., 2018; Celikyilmaz et al., 2018; Narayan et al., 2018c; Chen and Bansal, 2018;

[Peyrard and Gurevych, 2018](#)) agree on two broad quality definitions: *coverage* determines how much of the salient content of the source document is captured in the summary, and *informativeness*, how much of the content captured in the summary is salient with regards to the original document. These measures correspond to “*recall*” and “*precision*” metrics respectively in [Table 2.2](#), notions that are commonly used in information retrieval and information extraction literature. [Clarke and Lapata \(2010\)](#) proposed a question-answering based approach to improve the agreement among human evaluations for the quality of summary content, which was employed by [Narayan et al. \(2018c\)](#) and [Narayan et al. \(2018b\)](#) (QA in [Table 2.2](#)). In this approach, questions were created first from the reference summary and then the system summaries were judged with regards to whether they enabled humans to answer those questions correctly. [ShafeiBavani et al. \(2018\)](#), on the other hand, used the Pyramid method ([Nenkova and Passonneau, 2004](#)) which requires summaries to be annotated by experts for salient information. A similar evaluation approach is the factoids analysis by [Teufel and Van Halteren \(2004\)](#) which evaluates the system summary against factoids, a representation based on atomic units of information, that are extracted from multiple gold summaries. However, as in the case of the “Pyramid” method, extracting factoids requires experts annotators. Finally, a small number of works evaluates the “Correctness” ([Chen and Bansal, 2018](#); [Li et al., 2018b](#); [Chen and Bansal, 2018](#)) of the summary, similar to fact checking ([Vlachos and Riedel, 2014](#)), which can be a challenging task in its own right.

The linguistic quality of a summary encompasses many different qualities such as fluency, grammatically, readability, formatting, naturalness and coherence. Most works

uses a single human judgment to capture all linguistic qualities of the summary [Hsu et al. \(2018\)](#); [Kryściński et al. \(2018\)](#); [Narayan et al. \(2018c\)](#); [Song et al. \(2018\)](#); [Groschwitz et al. \(2018\)](#); we group them under “Fluency” in Table 2.2 with an exception of “Clarity” which was evaluated in the DUC evaluation campaigns ([Dang, 2005](#)). The “Clarity” metric puts emphasis on easy identification of noun and pronoun phrases in the summary which is a different dimension from “Fluency”, as a summary may be fluent but difficult to be understood due to poor clarity.

Absolute vs Relative Summary Ranking. In relative assessment of summarization, annotators are shown two or more summaries and are asked to rank them according to the dimension at the question ([Yang et al., 2017](#); [Chen and Bansal, 2018](#); [Narayan et al., 2018a](#); [Groschwitz et al., 2018](#); [Krishna and Srinivasan, 2018](#)). The relative assessment is often done using the paired comparison ([Thurstone, 1994](#)) or the best-worst scaling ([Woodworth and G, 1991](#); [Louviere et al., 2015](#)), to improve the inter-annotator agreement. On the other hand, absolute assessment of summarization ([Li et al., 2018b](#); [Song et al., 2018](#); [Kryściński et al., 2018](#); [Hsu et al., 2018](#); [Hardy and Vlachos, 2018](#)) is often done using the Likert rating scale ([Likert, 1932](#)) where a summary is assessed on a numerical scale. Absolute assessment was also employed in combination with the question answering approach for content evaluation ([Narayan et al., 2018c](#); [Mendes et al., 2019](#)). Both approaches, relative ranking and absolute assessment, have been investigated extensively in Machine Translation ([Bojar et al., 2016, 2017](#)). Absolute assessment correlates

highly with the relative assessment without the bias introduced by having a simultaneous assessment of several models (Bojar et al., 2011).

Choice of Reference. The most convenient way to evaluate a system summary is to assess it against the reference summary (Celikyilmaz et al., 2018; Yang et al., 2017; Peyrard and Gurevych, 2018), as this typically requires less effort than reading the source document. The question answering approach of Narayan et al. (2018c,b) also falls in this category, as the questions were written using the reference summary. However, summarization datasets are limited to a single reference summary per document (Sandhaus, 2008; Hermann et al., 2015; Grusky et al., 2018; Narayan et al., 2018c) thus evaluations using them are prone to reference bias (Louis and Nenkova, 2013), also a known issue in machine translation evaluation (Fomicheva and Specia, 2016). A way to circumvent this issue is to evaluate against the source document (Song et al., 2018; Narayan et al., 2018a; Hsu et al., 2018; Kryściński et al., 2018), asking judges to assess the summary after reading the source document. However this requires more effort and is known to lead to low inter-annotator agreement (Nenkova and Passonneau, 2004).

2.6 Datasets for Summarization

There are many datasets that are used for summarization purposes. Many of them come from shared tasks while others crawled from the Internet.

DUC and TAC Datasets DUC and TAC datasets comprises of documents and their respective summaries sampled from various newswire dataset. This datasets cover many different summarization tasks, such as multi-document, single document and others.

Proxy Report section of the AMR Dataset The proxy Report section of the AMR dataset comprises various newswire datasets that are also annotated with AMR parses. The size of this dataset is however quite small and isn't fit for a seq2seq summarization approach. However this dataset is useful when one wants to build an AMR summarization ([Liu et al., 2015](#)).

eXtreme SUMmarization (XSUM) XSUM dataset ([Narayan et al., 2018c](#)) is a newswire article-summary pairs collected from BBC news articles. XSUM dataset contains higher level of abstractiveness compared to other datasets.

Gigaword ([Graff et al., 2003b](#)) The English Gigaword is a large collection of newswire collected from different sources. It contains document and headline pairs which can be used for a headline generator or summarization task.

The New York Times ([Sandhaus, 2008](#)) The New York Times (NYT) is a collection of 1.8 millions article written and published by the NYT. Each article comes with an abstractive summary written the journalists of the NYT.

CNN / DailyMail ([Hermann et al., 2015](#)) The CNN / DailyMail datasets contain newswire articles collected from CNN and DailyMail news. The articles have higher rate of extractive characteristic than abstractive since most of the summaries are using the lead lines of the articles.

WikiHow Wikihow dataset ([Koupae and Wang, 2018](#)) is a highly-abstractive summarization dataset that comes from the Wikihow website⁵. The summary comes from the first line of each paragraph in the article.

Reddit-TIFU Reddit-TIFU dataset ([Kim et al., 2018](#)) is also a highly abstractive summarization dataset that comes from Reddit specifically the TIFU subreddit. It comprises of users' posts as the source document and the TL;DR (Too Long; Don't Read) section as the target summary.

Table 2.3 summarizes all previously mentioned datasets' characteristics.

2.7 Conclusion and Final Remarks

In this chapter we have presented a survey of deep learning approaches, pre-trained language models, automatic summarization and its evaluation, and guidance mechanism.

The following is the recap of this chapter:

⁵<https://www.wikihow.com/>

1. We have discussed different approaches in Deep Learning approaches, i.e.: RNN, CNN and Transformer model. For each architecture, we have described the sequence-to-sequence (seq2seq) model that is used in abstractive summarization setting. We have also discussed pre-trained language model approach with the focus of BART which is a seq2seq pre-trained language model.
2. We have surveyed the landscape of automatic summarization and its different types. We have shown that seq2seq deep learning and pre-trained language models have substantially improved abstractive summarization however problems still persist in the model's output such as hallucination and others.
3. We have discussed different approaches in solving said problems in abstractive summarization using guidance mechanism. We observed that recent approaches in guiding mechanism that use structural information requires re-training the dataset every time the guidance changes.
4. We have surveyed different approaches in automatic and manual evaluation. We observed that in the current large dataset, the reference-bias issue is impacting the quality of evaluation.

Systems	No Manual Eval	Pyramid	QA	Correctness	Fluency	Clarity	Recall	Precision	Absolute	Relative	With Reference	With Document	With Ref. & Doc.
		See et al. (2017)	✓										
Lewis et al. (2020)	✓												
Lin et al. (2018)	✓												
Cohan et al. (2018)	✓												
Liao et al. (2018)	✓												
Kedzie et al. (2018)	✓												
Amplayo et al. (2018)	✓												
Jadhav and Rajan (2018)	✓												
Li et al. (2018a)	✓												
Pasunuru and Bansal (2018)	✓												
Cao et al. (2018)	✓												
Sakaue et al. (2018)	✓												
Yang et al. (2017)	✓							✓		✓	✓		
Celikyilmaz et al. (2018)								✓	✓	✓	✓	✓	
Chen and Bansal (2018)				✓				✓	✓	✓			✓
Groschwitz et al. (2018)					✓			✓		✓			✓
Hardy and Vlachos (2018)					✓				✓				
Hsu et al. (2018)					✓			✓	✓			✓	
Krishna and Srinivasan (2018)								✓		✓		✓	
Kryściński et al. (2018)					✓			✓	✓			✓	
Li et al. (2018b)				✓					✓				
Narayan et al. (2018a)					✓					✓		✓	
Narayan et al. (2018c)			✓		✓			✓	✓	✓	✓	✓	
Narayan et al. (2018b)			✓		✓			✓	✓	✓	✓	✓	
Peyrard and Gurevych (2018)								✓	✓		✓		
ShafieiBavani et al. (2018)		✓											
Song et al. (2018)				✓	✓			✓	✓			✓	
Zhang et al. (2020)								✓	✓		✓		
Dou et al. (2020)				✓					✓		✓		
Narayan et al. (2021)				✓						✓	✓		
HIGHRES (ours)					✓	✓	✓	✓	✓			✓	

TABLE 2.2: Overview of manual evaluations conducted in recent summarization systems. We categorize them in four dimensions: the first column presents papers that do not report on human evaluation; the second column identifies matrices used for evaluating content (“*Pyramid*”, “*QA*”, “*Correctness*”, “*Recall*” and “*Precision*”) and quality (“*Clarity*”, “*Fluency*”) of summaries; the third column focuses if the system ranking reported by humans on content evaluation were “*Absolute*” or “*Relative*”; and finally, the fourth column evaluates if summaries were evaluated against the input document (“*With Document*”), the reference summary (“*With Reference*”) or both (“*With Ref. & Doc.*”).

TABLE 2.3: Various dataset for summarization purpose

Corpus	Type	Size
DUC 2001 - 2002	Single Document and multi-document summarization	30 topics
DUC 2003 - 2004	Single Document and Multi-document summarization	60 topics
DUC 2005 - 2006	Focus Query summarization in Question Answering style	25 - 50 topics
DUC 2007, TAC 2008 - 2011	Focus Query summarization and Multi-Document Summarization	44 topics in 5 categories with 20 documents each
PROXY AMR	Single Document summarization	8252 sentences with gold annotated AMR
Gigaword	Single Document summarization	10 Million documents (4 billion words)
LDC2012T21	Single Document summarization	1.8 Million articles (more than 650 thousands summaries)
The New York Times Annotated Corpus	Single Document summarization	
LDC2008T19	Single Document summarization	399,147 articles
Extreme Summarization (XSUM)	Single Document summarization	
CNN-DM	Single Document summarization	907,179 articles
Wikipedia	Single Document summarization	230,843 articles
Reddit-TIFU	Single Document summarization	79,949 TIFU-short and 42,984 TIFU-long

Chapter 3

HighRES: Highlight-based Reference-less Evaluation of Summarization

In this chapter we propose a novel approach for manual evaluation, HIGHLIGHT-based Reference-less Evaluation of document Summarization (HIGHRES), in which a summary is assessed against the source document via manually highlighted salient content in the latter.

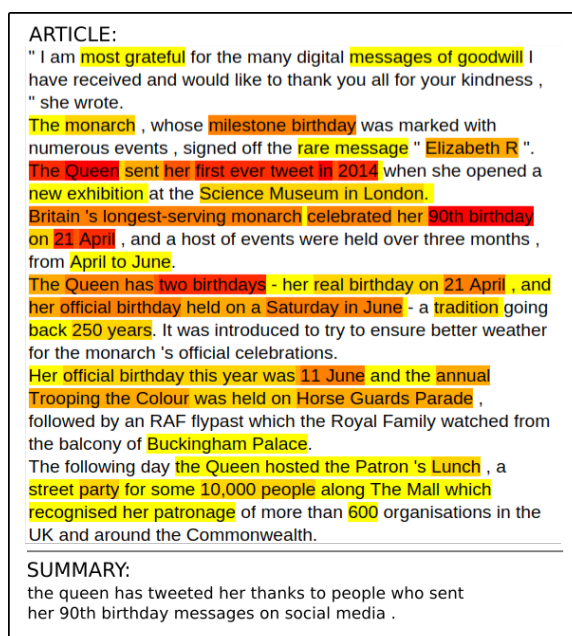


FIGURE 3.1: Highlight-based evaluation of summaries. Annotators to evaluate a summary (bottom) against the highlighted source document (top) presented with a heat map marking the salient content in the document; the darker the colour, the more annotators deemed the highlighted text salient.

3.1 Introduction

Progress in automatic summarization is determined by assessing how good a system produced summary. A good system summary must express salient information of the source document in a fluent and succinct manner. This is a difficult task as there can be multiple equally good summaries for the same source document as not all salient information can fit in a given summary length and also because human annotators are subject to a certain degree of subjectivity when creating a summary ([Harman and Over, 2004](#)).

One solution to the above problem is to provide multiple reference summaries for a given document as such we can give more weight to information that occurs in multiple

references. However, prominent datasets (Hermann et al., 2015; Narayan et al., 2018c; Koupae and Wang, 2018; Kim et al., 2018) only have a single reference summary available for each source document, as obtaining multiple ones increases dataset creation cost. Because of these, system evaluations that are using these datasets are likely to exhibit reference bias (Louis and Nenkova, 2013; Fomicheva and Specia, 2016) where good summaries that contain salient information different from the single reference are penalized.

Notable works in summarization evaluation such as Pyramid (Nenkova and Passonneau, 2004) and ROUGE (Lin, 2004) were created with the idea of multiple summaries. Another evaluation approach such as Narayan et al. (2018c,b) used question-answering to determine a system summary quality. All these approaches exhibit reference bias therefore they are not suitable for said datasets that only have a single reference summary.

We proposed a solution for the reference-bias problem by doing system summary evaluation directly against highlighted source document called **HIGHLIGHT-based Reference-less Evaluation of document Summarization (HIGHRES)**. We focus on manual evaluation since automatic measures are unlikely to be sufficient to measure performance in summarization (Schluter, 2017).

3.2 HighRES

Our novel highlight-based reference-less evaluation does not suffer from reference bias as a summary is assessed against the source document with manually highlighted salient

content. These highlights are crowd-sourced effectively without the need of expert annotators as required by the Pyramid method or to generate reference summaries. Our approach improves over the “Correctness” or “Fluency” only measure for summarization by taking salience into account. Finally, the assessment of summaries against the document with highlighted pertinent content facilitates an absolute evaluation of summaries with the high inter-annotator agreement.

Our evaluation framework comprises three main components: document highlight annotation, highlight-based content evaluation, and clarity and fluency evaluation. The second component, which evaluates the notions of “Precision” and “Recall” requires the highlights from the first one to be conducted. However, the highlight annotation needs to happen only once per document, and it can be reused to evaluate many system summaries, unlike the Pyramid approach that requires additional expert annotation for every system summary being evaluated. The third component is independent of the others and can be run in isolation. In all components, we employ crowd-workers as human judges and implement appropriate sanity checking mechanisms to ensure good quality judgements. Finally, we present an extended version of ROUGE that utilizes the highlights to evaluate system summaries against the document; this demonstrates another use of the highlights for summarization evaluation.

3.2.1 Highlight Annotation

In this part, we ask human judges to read the source document and then highlight words or phrases that are considered salient. Each judge is allowed to highlight parts of the text at any granularity, from single words to complete sentences or even paragraphs. However, we enforce a limit in the number of words to \mathcal{K} that can be highlighted in total by a judge in a document, corresponding to the length of the summary expected. The user interface for highlighting annotation can be seen in Figure 3.2.

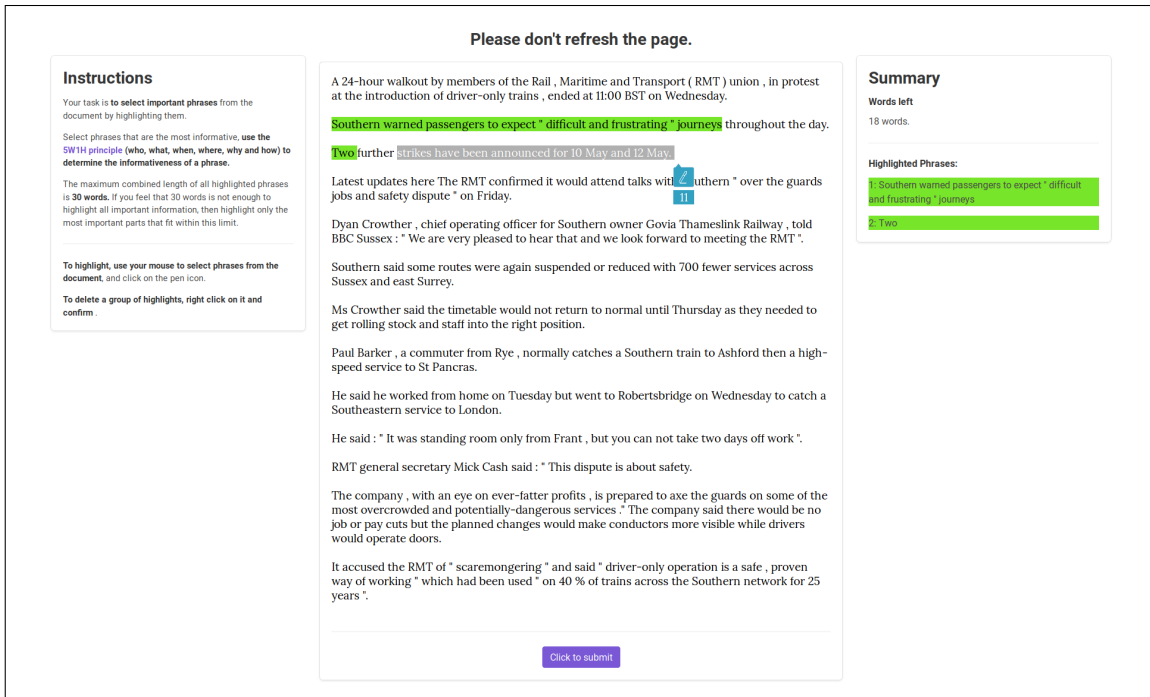
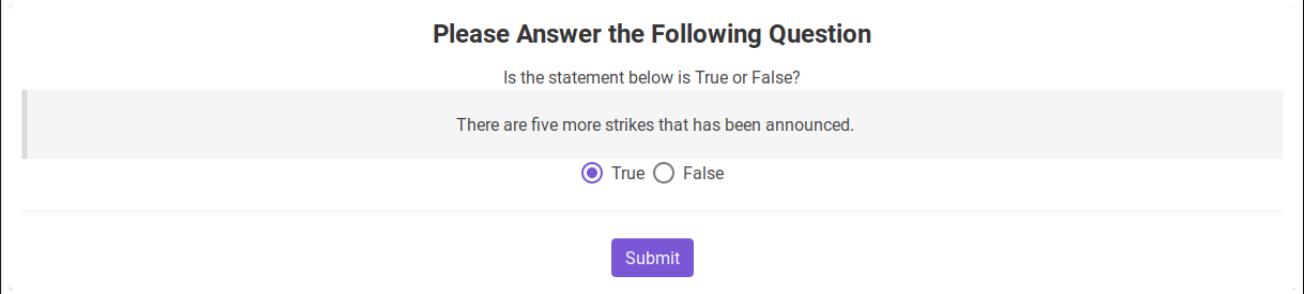


FIGURE 3.2: The UI for highlight annotation. Judges are given an article and asked to highlight words or phrases that are important in the article.

By employing multiple judges per document who are restricted in the amount of text that can be highlighted we expect to have a more diverse and focused highlight from multiple judges who cover different viewpoints of the article.

To ensure that each highlight is reliable, we performed a sanity check at the end of the task where we ask the judges to answer a True/False question based on the article. We rejected all annotations that failed to correctly answer the sanity check question. The user interface for highlight annotation sanity checking can be seen in Figure 3.3.



Please Answer the Following Question

Is the statement below is True or False?

There are five more strikes that has been announced.

True False

FIGURE 3.3: The sanity checking question at the end of the annotation task.

3.2.2 Highlight-based Content Evaluation

In this component, we present human judges a document that has been highlighted using heatmap coloring and a summary to assess (see Figure 3.1 for an example). We ask our judges to assess the summary for (i) ‘*All important information is present in the summary*’ and (ii) ‘*Only important information is in the summary.*’ The first one is the recall (content coverage) measure and the second, the precision (informativeness) measure. All the ratings were collected on a 1-100 Likert scale (Likert, 1932). For baselines, we also did a similar content evaluation but without highlights and reference only (comparing directly against the reference summary).

The user interface for content evaluation with highlights, without highlights and reference only can be seen in Figure 3.4, Figure 3.5 and Figure 3.6

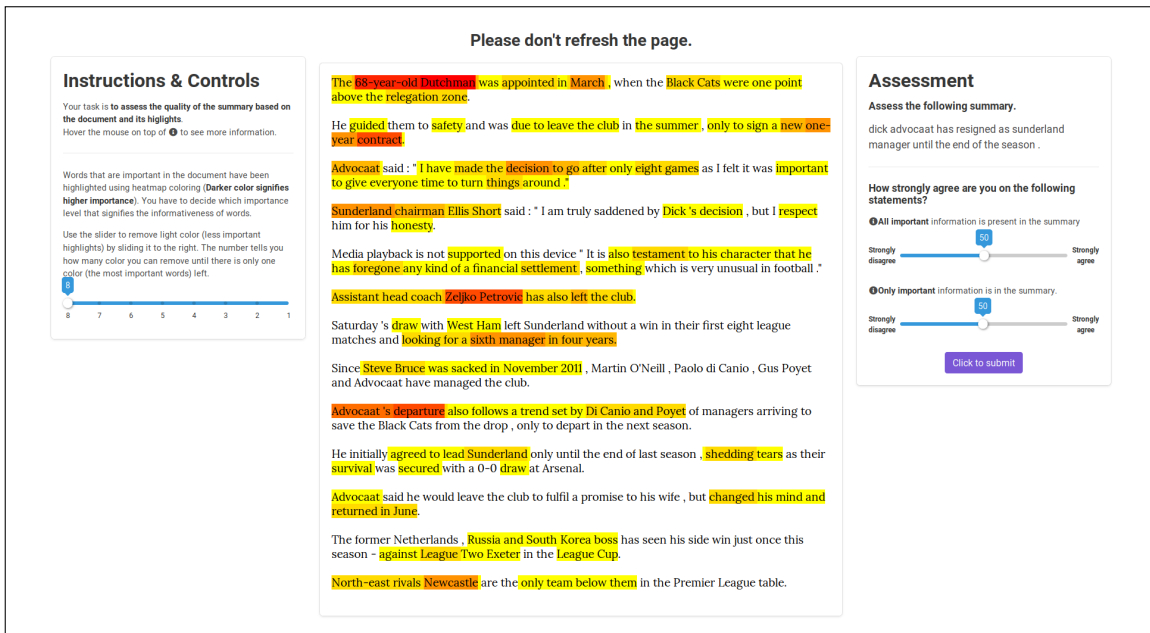


FIGURE 3.4: The UI for content evaluation with highlights. Judges are given an article with important words highlighted using a heat map. Judges can also remove less important highlight color by sliding the scroller at the left of the page. At the right of the page, judges give the recall and precision assessment by sliding the scroller from 1 to 100 based on the given summary quality.

As with the highlight annotation, we also performed the same form of a sanity check as the one in the highlight annotation task. The user interface for content evaluation sanity checking is similar to Figure 3.3.

3.2.3 Clarity and Fluency Evaluations

In this part, we give the judges only the summary and ask them to rate it on clarity and fluency. For *clarity*, each judge is asked whether the summary is easy to be understood, i.e. there should be no difficulties in identifying the referents of the noun phrases (every noun/place/event should be well-specified) or understanding the meaning of the sentence.

Please don't refresh the page.

Instructions & Controls

Your task is to assess the quality of the summary based on the document.

Labour MP for Bassetlaw, John Mann, a Leave campaigner, said people voted to leave because of immigration, zero-hour contracts and job prospects and said a "divide in Britain" had been exposed.

Mansfield voted most strongly to leave, with 70.9% backing Brexit.

Rushcliffe, which includes the towns of West Bridgford and Bingham, was the only area to vote for Remain.

It saw the East Midlands' highest turnout.

Meanwhile, the turnout in Nottingham was the fifth lowest in the UK at 61.8%.

Leave won by a tiny margin of just over 2,000 votes in the city.

Mr Mann said his party was "somewhat out of touch".

"With the middle classes largely voting remain because they see it as benefiting them and the working classes largely voting to leave because it dis-benefits them - that's the divide in Britain," he said.

Latest reaction and updates from Nottinghamshire Like large parts of England, Nottinghamshire overwhelmingly voted to leave the European Union.

The margin of victory in Bassetlaw, Ashfield and Mansfield was huge, with less than a third of people voting remain.

Arguably the biggest surprise came in Nottingham, which narrowly backed Brexit.

Affluent Rushcliffe was the only area to vote Remain.

Overall Nottinghamshire voted 57.9% for Leave and 42.1% for Remain.

Conservative Anna Soubry, the MP for Broxtowe and a Remain campaigner, tweeted it was "a dreadful decision".

"People like me were told you're scaremongering, we do n't want to listen to the experts," she said.

"All that has been unfortunately proved to be accurate.

We have made a very, very, very bad mistake." Labour MP for Nottingham North, Graham Allen, said David Cameron had "gambled with Britain's future" by calling for a referendum, saying people voted to leave "in protest" at the current government.

Turnout was 81.5% in Rushcliffe - the highest in the East Midlands and the only council area to vote Remain.

The vote was close elsewhere, including in Nottingham, where the split was 50.8% Leave, 49.2% Remain.

Alice, a caller to BBC Radio Nottingham from the Carrington area of the city, said she felt "frightened of the future".

"It potentially gives a mandate for a lot of prejudice against people who have immigrated here, whether from the EU or elsewhere," she said.

Assessment

Assess the following summary.

labour has voted to leave the european union after voters voted to leave the eu.

How strongly agree are you on the following statements?

All important information is present in the summary

Strongly disagree 50 Strongly agree

Only important information is in the summary.

Strongly disagree 50 Strongly agree

[Click to submit](#)

FIGURE 3.5: The UI for content evaluation without highlight. At the right of the page, judges give the recall and precision assessment by sliding the scroller from 1 to 100 based on the given summary quality.

For *fluency*, each judge is asked whether the summary sounds natural and has no grammatical problems. While fluency is often evaluated in summarization work, clarity, while first introduced in DUC evaluations, has largely been ignored in manual evaluation, despite that it captures a different dimension of summarization quality. The user interface for clarity and fluency evaluations can be seen in Figure 3.7 and Figure 3.8.

To ensure that the judgments for clarity and fluency are not affected by each other (poor fluency can affect clarity, but a summary can have perfect fluency but low clarity), we evaluate each metric separately. We ask the judges to evaluate multiple summaries

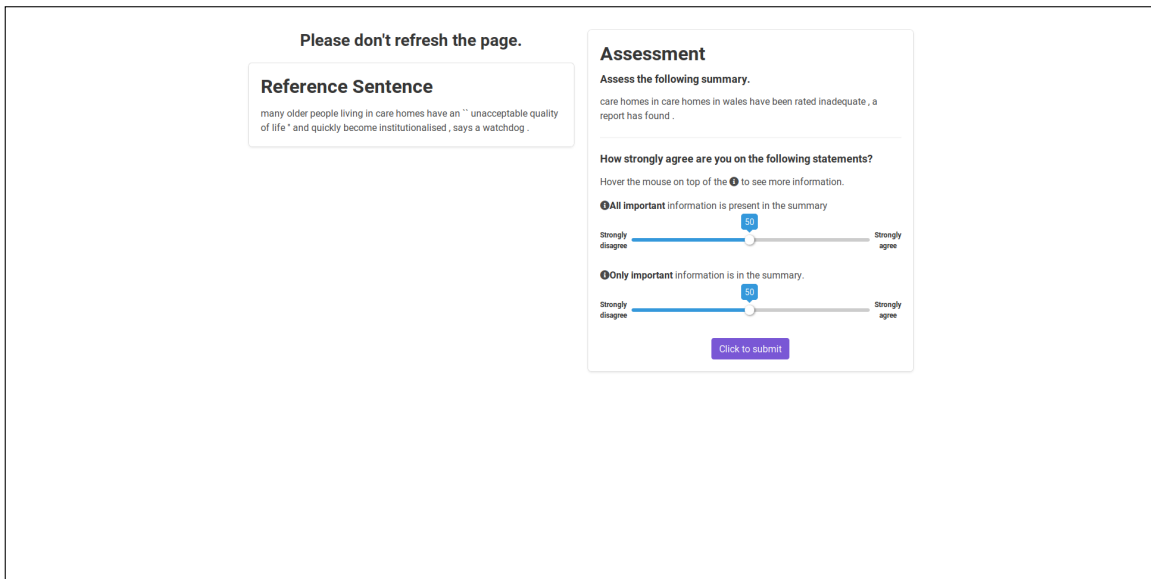


FIGURE 3.6: The UI for content evaluation using a reference summary as a comparison. At the right of the page, judges give the recall and precision assessment by sliding the scroller from 1 to 100 based on the given summary quality.

per task with each dimension on its own screen. For sanity checking, we insert three fakes summaries with different qualities (good, mediocre and bad summaries). We reject results that failed to pass this criterion: $bad < mediocre < good$.

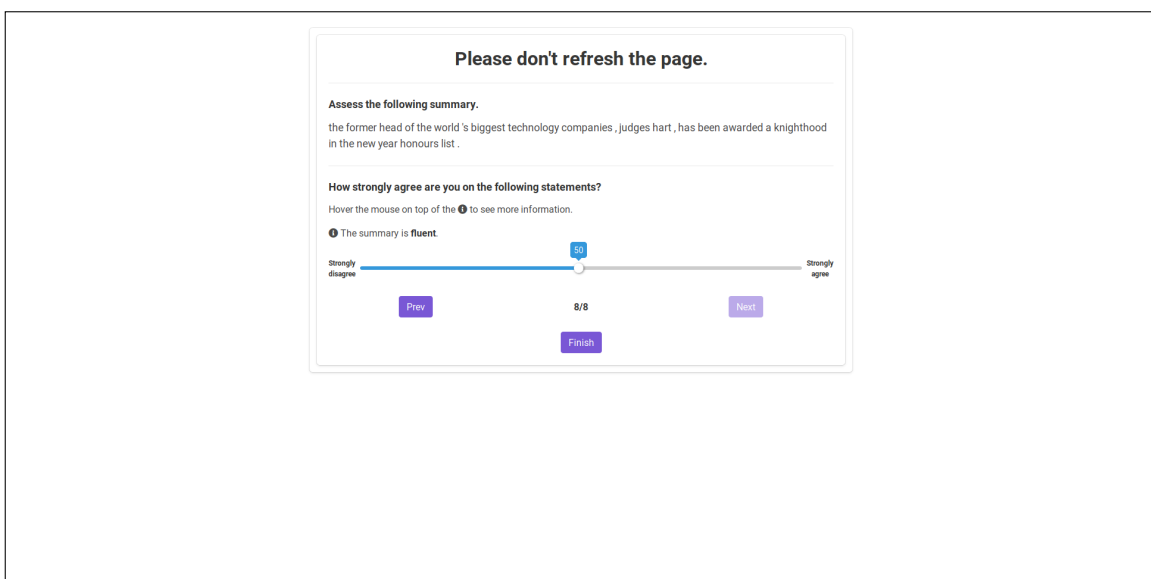


FIGURE 3.7: The UI for fluency evaluation. Judges are given a number of summaries which can be switched by pressing the ‘Prev’ or ‘Next’ button. To give an assessment, there is a scroller from 1 to 100.

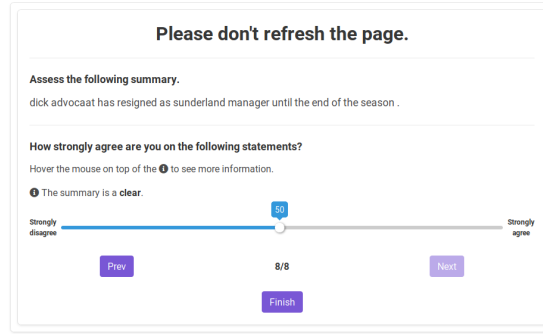


FIGURE 3.8: The UI for clarity evaluation. Judges are given a number of summaries which can be switched by pressing the ‘Prev’ or ‘Next’ button. To give an assessment, there is a scroller from 1 to 100.

3.2.4 Highlight-based ROUGE Evaluation

Our Highlight-based ROUGE (we refer to it as HROUGE) formulation is similar to the original ROUGE with the difference that the n -grams are weighted by the number of times they were highlighted. One benefit of HROUGE is that it introduces saliency into the calculation without being reference-based as in ROUGE. Implicitly HROUGE considers multiple summaries as the highlights are obtained from multiple workers.

Given a document \mathcal{D} as a sequence of m tokens $\{w_1, \dots, w_m\}$, annotated with \mathcal{N} highlights, we define the weight $\beta_g^n \in [0, 1]$ for an n -gram g as:

$$\beta_g^n = \frac{\sum_{i=1}^{m-(n-1)} \left[\frac{\sum_{j=i}^{i+n-1} \frac{\text{NumH}(w_j)}{\mathcal{N}}}{n} \right]_{w_{i:i+n-1}=g}}{\sum_{i=1}^{m-(n-1)} [1]_{w_{i:i+n-1}=g}} \quad (3.1)$$

where, $[x]_y$ is an indicator function which returns x if y is true and 0, otherwise. $\text{NumH}(w_j) = \sum_{k=1}^{\mathcal{N}} \frac{\text{len}(H_k)}{\mathcal{K}} [1]_{w_j \in H_k}$ is a function which returns the number of times word w_j is highlighted by the annotators out of \mathcal{N} times weighted by the lengths of their highlights; H_k is the highlighted text by the k th annotator and \mathcal{K} is the maximum allowed length of the highlighted text (see Section 3.2.1). $\text{NumH}(w_j)$ gives less importance to annotators with highlights with few words. In principle, if an n -gram is highlighted by every crowd-worker and the length of the highlight of each crowd-worker is \mathcal{K} , the n -gram g will have a maximum weight of $\beta_g^n = 1$. The HROUGE scores for a summary \mathcal{S} can then be defined as:

$$\text{HR}_{\text{rec}}^n = \frac{\sum_{g \in n\text{-gram}(\mathcal{S})} \beta_g^n \text{count}(g, \mathcal{D} \cap \mathcal{S})}{\sum_{g \in n\text{-gram}(\mathcal{D})} \beta_g^n \text{count}(g, \mathcal{D})} \quad (3.2)$$

$$\text{HR}_{\text{pre}}^n = \frac{\sum_{g \in n\text{-gram}(\mathcal{S})} \beta_g^n \text{count}(g, \mathcal{D} \cap \mathcal{S})}{\sum_{g \in n\text{-gram}(\mathcal{S})} \text{count}(g, \mathcal{S})} \quad (3.3)$$

HR_{rec}^n and HR_{pre}^n are the HROUGE recall and precision scores; $\text{count}(g, \mathcal{X})$ is the maximum number of n -gram g occurring in the text \mathcal{X} . There is no weighting in the denominator of precision ($\beta_g^n = 1$) as if we weighted according to the highlights, words in the summary that are not highlighted in the original document would be ignored. This would result in HR_{pre}^n not penalizing summaries for containing words that are likely to be irrelevant as they do not appear in the highlights of the document. It is important to note HROUGE has an important limitation in that it penalizes abstractive summaries that do not reuse words from the original document. This is similar to ROUGE penalizing

summaries for not reusing words from the reference summaries, however, the highlights implicitly consider multiple references.

3.3 Summarization Dataset and Models

We use the extreme summarization dataset (XSUM, [Narayan et al., 2018c](#))¹ which comprises BBC articles paired with their single-sentence summaries, provided by the journalists writing the articles. The summary in the XSUM dataset demonstrates a larger number of novel n -grams compared to other popular datasets such as CNN/DailyMail ([Hermann et al., 2015](#)) or NY Times ([Sandhaus, 2008](#)). This makes the dataset suitable for our experiment since the more abstractive nature of the summary renders automatic methods such as ROUGE less accurate as they rely on string matching, and thus calls for human evaluation for more accurate system comparisons. Following [Narayan et al. \(2018c\)](#), we didn't use the whole test set portion but sampled 50 articles from it for our highlight-based evaluation.

We assessed summaries from two abstractive summarization systems using our highlight-based evaluation:

1. Pointer-Generator model (PTGEN) introduced by [See et al. \(2017\)](#) is an RNN-based abstractive system that allows copying words from the source text.

¹<https://github.com/EdinburghNLP/XSum>

2. Topic-aware Convolutional Sequence to Sequence model (TCONVS2S) introduced by Narayan et al. (2018c) is an abstractive model which is conditioned on the article’s topics and based entirely on Convolutional Neural Networks

We used the pre-trained models² provided by the authors to obtain summaries from both systems for the documents in our test set.

3.4 Experiments

All of our experiments were done using the Amazon Mechanical Turk platform. We developed three types of Human Intelligence Tasks (HITs): highlight annotation, highlight-based content evaluation, and fluency and clarity evaluation. In addition, we elicited human judgments for content evaluation in two more ways: we assessed system summaries against the original document (without highlights) and against the reference summary. The latter two experiments were intended as the comparison for our proposed highlight-based content evaluation.

3.4.1 Highlight Annotation

We collected highlight annotations from 10 different participants for each of the 50 articles. For each annotation, we set \mathcal{K} , the maximum number of words to highlight, to 30.

Our choice reflects the average length (24 words) of reference summaries in the XSUM

²Both models were trained using the standard cross-entropy loss to maximize the likelihood of the reference summary given the document.

dataset. To facilitate the annotation of BBC news articles with highlights, we asked our participants to adapt the 5W1H (Who, What, When, Where, Why and How) principle (Robertson, 1946) that is a common practice in journalism. The participants however were not obliged to follow this principle and were free to highlight content as they deem fit.

The resulting annotation exhibits a substantial amount of variance, confirming the intuition that different participants are not expected to agree entirely on what is salient in a document. On average, the union of the highlights from 10 annotators covered 38.21% per article and 33.77% of the highlights occurred in the second half of the article. This shows that the judges did not focus only on the beginning of the documents but annotated all across the document.

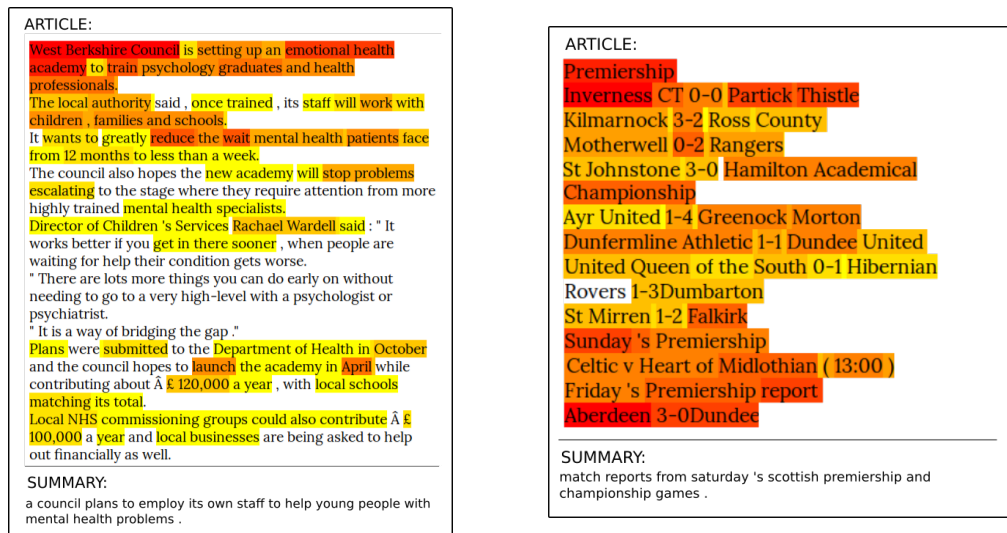


FIGURE 3.9: Highlight annotation for the documents with the highest (left) and lowest (right) agreement. We also show their reference summaries at the bottom.

Using Fleiss Kappa (Fleiss, 1971) on the binary labels provided by each judge on each

word (highlighted or not) we obtained an average agreement of 0.19 for the 50 articles considered. Article with the highest agreement (0.32) has more focused highlights, whereas the article with the lowest agreement (0.04) has highlights spread all over (both articles can be seen in Figure 3.9). Interestingly, the reference summary on the highest agreement article appears to be more informative of its content for which the annotator agreement is high; the reference summary on the lowest agreement article is more indicative, i.e., it describes the document rather than directly presenting the information it contains. These results confirm that the annotation behaviour originates from the nature of the document and the summary it requires and validates our highlight annotation setup.

Model	Highlight-based		Non Highlight-based		Reference-based	
	Prec	Rec	Prec	Rec	Prec	Rec
TCONVS2S	57.42	49.95	52.55	41.04	46.75	36.45
PTGEN	50.94	44.41	48.57	39.21	44.24	38.24
Reference	67.90	56.83	66.01	52.45	—	—

TABLE 3.1: Results of content evaluation of summaries against documents with highlights, documents without highlights and reference summaries.

Model	Highlight-based		Non Highlight-based	
	Prec	Rec	Prec	Rec
TCONVS2S	0.67	0.80	0.75	0.83
PTGEN	0.73	0.86	0.73	0.90
Reference	0.49	0.63	0.48	0.67

TABLE 3.2: Coefficient of variation (lower is better) for evaluating summaries against documents with and without highlights.

3.4.2 Content Evaluation of Summaries

We assessed the summaries against (i) documents with highlights (Highlight-based), (ii) original documents without highlights (Non Highlight-based) and (iii) reference summaries (Reference-based). For each setup, we collected judgments from 3 different participants for each model summary. Table 3.1 and 3.2 presents our results.

Both the highlight-based and non-highlight based assessment of summaries agrees on the ranking among TCONVS2S, PTGEN and Reference. Perhaps unsurprisingly human-authored summaries were considered best, whereas, TCONVS2S was ranked 2nd, followed by PTGEN. However, the performance difference in TCONVS2S and PTGEN is greatly amplified when they are evaluated against documents with highlights (6.48 and 5.54 Precision and Recall points) compared to when evaluated against the original documents (3.98 and 1.83 Precision and Recall points). The performance difference is lowest when they are evaluated against the reference summary (2.51 and -1.79 Precision and Recall points). The superiority of TCONVS2S is expected; TCONVS2S is better than PTGEN for recognizing pertinent content and generating informative summaries due to its ability to represent high-level document knowledge in terms of topics and long-range dependencies (Narayan et al., 2018c).

We further measured the agreement among the judges using the coefficient of variation (Everitt, 2006) from the aggregated results. It is defined as the ratio between the sample standard deviation and the sample mean. It is a scale-free metric, i.e. its results are comparable across measurements of different magnitude. Since, our sample size is small

Model	Fluency	Clarity
TCONVS2S	69.51	67.19
PTGEN	55.24	52.49
Reference	77.03	75.83

TABLE 3.3: Mean "Fluency" and "Clarity" scores for TCONVS2S , PTGEN and Reference summaries. All the ratings were collected on a 1-100 Likert scale.

(3 judgements per summary), we use the unbiased version (Sokal and Rohlf, 1995) as $cv = (1 + \frac{1}{4n})\frac{\sigma}{\bar{x}}$, where σ is the standard deviation, n is the number of sample, and \bar{x} is the mean.

We found that the highlight-based assessment, in general, has lower variation among judges than the non-highlight based or reference-based assessment. The assessment of TCONVS2S summaries achieves 0.67 and 0.80 of Precision and Recall cv points which are 0.08 and 0.03 points below when they are assessed against documents with no highlights, respectively. We see a similar pattern in Recall on the assessment of PTGEN summaries. Our results demonstrate that the highlight-based assessment of abstractive systems improve agreement among judges compared to when they are assessed against the documents without highlights or the reference summaries. The assessment of human-authored summaries does not seem to follow this trend, we report mixed results (0.49 vs 0.48 for precision and 0.63 vs 0.67 for recall) when they are evaluated with and without the highlights.

Model	Unigram		Bigram	
	Prec	Rec	Prec	Rec
ROUGE (Original document)				
TCONVS2S	77.17	4.20	26.12	1.21
PTGEN	77.09	4.99	28.75	1.64
Reference	73.65	4.42	22.42	1.17
HROUGE (Highlights from the document)				
TCONVS2S	7.94	5.42	3.30	2.11
PTGEN	7.90	6.46	3.37	2.64
Reference	7.31	5.73	2.39	1.84

TABLE 3.4: HROUGE-1 (unigram) and HROUGE-2 (bigram) precision, and recall scores for TCONVS2S , PTGEN and Reference summaries.

3.4.3 Clarity and Fluency Evaluation

Table 3.3 shows the results of our fluency and clarity evaluations. Similar to our highlight-based content evaluation, human-authored summaries were considered best, whereas TCONVS2S was ranked 2nd followed by PTGEN, on both measures. The Pearson correlation between fluency and clarity evaluation is 0.68 which shows a weak correlation; it confirms our hypothesis that the "clarity" captures different aspects from "fluency" and they should not be combined as it is commonly done.

3.4.4 Highlight-based ROUGE Evaluation

Table 3.4 presents our HROUGE results assessing TCONVS2S , PTGEN and Reference summaries with the highlights. To compare, we also report ROUGE results assessing these summaries against the original document without highlights. In the latter case, HROUGE becomes the standard ROUGE metric with $\beta_g^n = 1$ for all n -grams g .

Both ROUGE and HROUGE favour the method of copying content from the original document and penalizes abstractive methods, thus it is not surprising that PTGEN is superior to TCONVS2S, as the former has an explicit copy mechanism. The fact that PTGEN is better in terms of HROUGE is also evidence that the copying done by PTGEN selects salient content, thus confirming that the copying mechanism works as intended. When comparing the reference summaries against the original documents, both ROUGE and HROUGE confirm that the reference summaries are rather abstractive as reported by [Narayan et al. \(2018c\)](#), and they in fact score below the system summaries. Recall scores are very low in all cases which is expected, since the 10 highlights obtained per document or the documents themselves, taken together, are much longer than any of the summaries.

3.5 Qualitative Analysis

3.5.1 HighRES eliminates reference bias.

The example presented in [Figure 3.10](#) demonstrates how our highlight-based evaluation eliminates reference bias in summarization evaluation. In it, we can see that summaries generated by TCONVS2S and PTGEN are able to capture the essence of the document. For example, the words bolded by red color in TCONVS2S such as ‘met office’ and ‘yellow’ don’t come out in the reference summary but HIGHRES shows highlights on both words giving the judge more information. A reference-based evaluation would fail to give a

<p>ARTICLE:</p> <p>The yellow warning will remain in force until 11:00 on Sunday. Forecasters said showers accompanied by widespread sub-zero temperatures would see ice form on many untreated roads. Some snow is expected even at low levels in northern Scotland and other areas could see 2-3cm fall on higher ground. A Met office forecaster said : " Over northern Scotland showers will fall as snow to low levels. " Elsewhere within the warning area these showers will be turning increasingly wintry , with the main snow level down to between 100 and 200m by the end of the night. " Locally 2 or 3 cm of snow is possible above 200m . "</p>
<p>SUMMARY:</p> <p>Reference: a weather warning has been issued for most parts of scotland , with drivers urged to be aware of a risk of ice and snow .</p> <p>TCONVS2S: the met office has issued a yellow `` be aware " warning for snow in parts of northern scotland .</p> <p>PTGEN: forecasters have warned of severe thunderstorms across parts of scotland and scotland as snow is forecast to affect wintry weather .</p>

FIGURE 3.10: The highlighted article, reference summary, and summaries are generated by TCONVS2S and PTGEN. Words in red in the system summaries are highlighted in the article but do not appear in the reference.

reasonable score to these system summaries. The HIGHRES however, would enable the judges to better evaluate the summaries without any reference bias.

In Figure 3.9 we also show two examples of the highest and lowest agreement for highlight annotation. In the highest agreement example, it is shown that reference summary couldn't capture the most important words in the article while our highlights could. In the lowest agreement example, we have shown that our highlight annotation could identify an indicative article while reference alone couldn't.

3.5.2 Fluency vs Clarity.

Example in Table 3.5 shows disagreements between fluency and clarity scores for different summaries of the same article. From the example, we can see that the TCONVS2S

Model	Summary Text	Fluency	Clarity
TCONVS2S	dick advocaat has resigned as sunderland manager <i>until the end of the season</i> .	92.80	44.33
PTGEN	sunderland have appointed <i>former sunderland boss</i> dick advocaat as manager <i>at the end of the season</i> to sign a <i>new deal</i> .	41.33	6.00

TABLE 3.5: TCONVS2S and PTGEN showing a disagreement between fluency and clarity scores. We italicized words that are not clear in the summaries.

summary is fluent but is not easily understood in the context of ‘the duration of resignation’, while the PTGEN summary has word duplication which lowers the fluency and also lacking clarity due to several unclear words.

3.6 Conclusion

In this chapter, we introduced the HIGHlight-based Reference-less Evaluation Summarization (HIGHRES) framework for manual evaluation. The proposed framework avoids reference bias and provides absolute instead of ranked evaluation of the systems. We also performed crowd-sourced annotations to obtain a highlighted source document for the purpose of HIGHRESevaluation. This annotated dataset also gives more understanding of how humans select salient information from the source document.

Our experiments show that HIGHRES lowers the variability of the judges’ content assessment while helping expose the differences between systems. We also showed that by evaluating clarity we are able to capture a different dimension of summarization quality

that is not captured by the commonly used fluency. We believe that our highlight-based evaluation is an ideal setup of abstractive summarization for three reasons:

1. Highlights can be crowdsourced effectively without expert annotations.
2. HIGHRES avoids reference bias.
3. HIGHRES is not limited by n-gram overlap.

With regards to research questions, our experiment answers the following research question:

Can we devise a new manual evaluation that is not affected by the reference bias problem? We developed a new framework that addresses the reference bias problem which consists of a web application and annotated datasets where annotators can use it to annotate a new highlighted document or use the existing dataset to evaluate systems' summaries.

For future works, we would like to extend our annotation process to multiple new datasets.

Chapter 4

Guided Neural Language Generation of AMR for Summarization

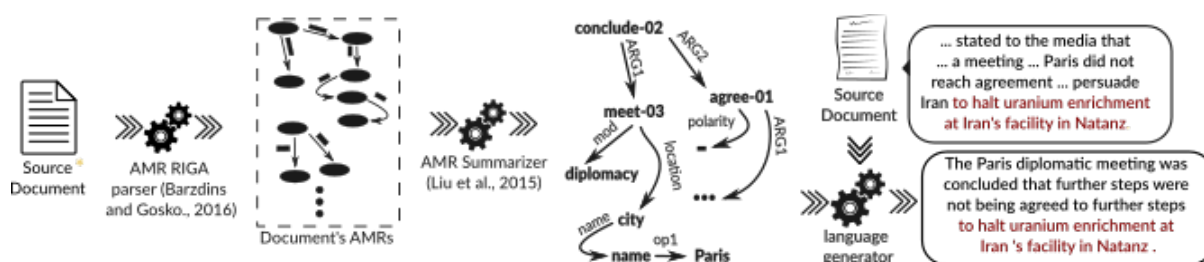


FIGURE 4.1: An Overview Diagram of the Guided NLG of AMR for summarization.

In this chapter, we propose a guided neural language generation of AMR for summarization. The overview diagram for our approach can be seen in Figure 4.1.

4.1 Introduction

Research (See et al., 2017; Chopra et al., 2016; Rush et al., 2015) in abstractive summarization has made progress with the neural encoder-decoder architecture. However, these models are often challenged when they were required to combine semantic information in order to generate a longer summary (Wiseman et al., 2017). The longer summary tends to display references error, incoherence and a lack of fidelity from the source document. Reference error means that the produced summary contains any entities or concepts that aren't found in the source document. Incoherence means that the produced summary lacks cohesion between its concepts. Finally, lack of fidelity means that the produced summary produced any information that is not in line with the facts in the source document. An example containing these errors can be seen in Table 4.1.

Model	Generated Summary
Gold	on 8 august 2008 russia conducted airstrikes on georgian targets .
Seq2seq (See et al., 2017)	the russian laboratory complex is a 90 - building campus and served as the location for russia 's secret biological weapons program in the soviet era of a moscow regional depository threaten moscow .

TABLE 4.1: Neural network model hallucination.

To address these shortcomings, we investigated the use of Abstract Meaning Representation (Banarescu et al., 2013, AMR) in an abstractive summarization task. Our

motivation is that AMR has the capability to capture the predicate-argument structure which can be utilized in an information aggregation process during summarization. However, the use of AMR also has its own shortcomings. While AMR is suitable for information aggregation, it ignores aspects of language such as tenses, grammatical numbers, etc., which are important for the natural language generation (NLG) stage that normally occurs at the end of the summarization process. Due to the lack of such information, approaches for NLG from AMR typically infer it from regularities in the training data (Pourdamghani et al., 2016; Konstas et al., 2017; Song et al., 2016; Flanigan et al., 2016). Due to this limitation, previous work on an AMR-based abstractive summarization (Liu et al., 2015) only generated bag-of-words from the summary AMR graph. We proposed an approach to guide the NLG stage in an AMR-based abstractive summarization using information from the source document (also called side information). Our objective is twofold:

1. To retrieve the information missing from AMR but needed for NLG
2. Improve the quality of the summary

We achieve our objectives in two stages:

1. Estimating the probability distribution of the side information.
2. Using it to guide a Luong et al. (2015)'s seq2seq model for NLG.

Our approach is evaluated using the Proxy Report from the AMR dataset (Knight et al., 2017, LDC2017T10) which contains manually annotated source documents and reference

summary AMR graphs. We built our work on top of Liu et al. (2015)’s work. Using our proposed guided AMR-to-text NLG, we improved summarization results using both gold standard AMR parses and parses obtained using the RIGA (Barzdins and Gosko, 2016) parser by 7.4 and 10.5 ROUGE-2 points respectively. Our model also outperformed a strong baseline seq2seq model (See et al., 2017) for summarization by 2 ROUGE-2 points.

4.2 Methodology

We first briefly describe the AMR-based summarization method of Liu et al. (2015) and then our guided NLG approach.

4.2.1 AMR-based summarization

In Liu et al. (2015)’s work, each of the sentences of the source document was parsed into an AMR graph and then combined into a source graph (see Chapter 2 for more information). Since they only generated bag-of-words (see Table 4.2) from the summary AMR graph, we extend their work to generate a fluent text in two different settings: *unguided* and *guided*.

To generate a fluent sentence from the BOW of Liu et al. (2015)’s system we used an AMR-to-text generator. There are two versions of the AMR-to-text generator that we used in this work. The first one is the baseline (unguided NLG from AMR) and our proposed model (the guided NLG from AMR).

 Generated AMR output

```
(x4 / state-01
  :ARG1 (x26 / reach-01
    :ARG1 (x14 / and
      :op2 (x15 / country :name (n/name :op1 "China" )))
    :ARG1 (x27 / agree-01
      :ARG1 (x32 / persuade-01
        :purpose (x35 / halt-01
          :ARG1 (x37 / enrich-01
            :ARG1 (x36 / uranium))
          :ARG0 (x33 / country :name (n/name :op1 "Iran" )))
        :ARG1 (x30 / action))
      :ARG0 (xap1 / person
        :ARG0-of (x16 / have-org-role-91
          :ARG1 (x19 / meet-03
            :time (x21 / date-entity :year 2008 :month 11 :day 13 ))))))))
```

 Bag of Word.

actions uranium officials and meeting stated halt 081113
 agreement china officials reach persuade iran enrichment

TABLE 4.2: System outputs of [Liu et al. \(2015\)](#). The bottom bag-of-words are generated from the top AMR tree.

4.2.2 Unguided NLG from AMR

Our baseline (unguided) is a standard seq2seq model with an attention mechanism ([Luong et al., 2015](#)) that consists of an encoder and decoder that takes the linearized summary AMR graph as the input and then generates a fluent text.

Before passing the linearized summary AMR graph, we run steps of preprocessing ([Van Noord and Bos, 2017](#)) on the AMR tree with the following steps:

1. Add extra space after all parentheses and remove the beginning and ending parentheses.
2. Remove all semantics identifiers from AMR concept nodes.
3. Delete the AMR variable.

The result of preprocessing can be seen in [Figure 4.2](#).

<pre>(m / material :mod (r / raw) :domain (o / opium) :ARG1-of (u / use-01 :ARG2 (p / make-01 :ARG1 (h / heroin) :ARG2 o)))</pre>	<pre>(material :mod (raw) :domain (opium) :ARG1-of (use-01 :ARG2 (make-01 :ARG1 (heroin) :ARG2 (opium))))</pre>
---	---

FIGURE 4.2: An example of the original AMR (left) and the variable-free AMR (right) displaying the meaning of *Opium is the raw material used to make heroin* (Van Noord and Bos, 2017).

After the preprocessing, we passed the linearized summary AMR graph into the seq2seq model. The encoder computes the hidden representation of the input, z_1, z_2, \dots, z_k . The decoder then generates a fluent text, y_1, y_2, \dots, y_m , using the conditional probability distribution $P_{s2s}(y_j|y_{<j}, z)$

4.2.3 Guided NLG from AMR

Our goal here is to improve the output of the unguided system in which tokens generation are sampled over the seq2seq probability distribution, P_{s2s} , by incorporating information from the source document. Since not all sentences in the source document will be used in generating the summary, we pruned the source document to a set of k sentences which have the highest similarity with the summary AMR graph. For graph-to-graph similarity comparison, we used the source document AMR parses and calculate the Longest Common Subsequence (LCS) between linearized AMR parses and the summary AMR graph. We keep the top- k sentences sorted by the LCS similarity. To distinguish this pruned document from the source document, we refer to the former as side information.

Our aim is to combine P_{s2s} with the probability distribution estimated using words in the side information, P_{side} , in order to score each word given its context during decoding. We estimated P_{side} as the linear interpolation of 2-gram to 4-gram probabilities in the form of

$$\begin{aligned} P_{side}(x_j|x_{j-3}^{j-1}) &= \lambda_3 P_{LM}(x_j|x_{j-3}^{j-1}) \\ &+ \lambda_2 P_{LM}(x_j|x_{j-2}^{j-1}) \\ &+ \lambda_1 P_{LM}(x_j|x_{j-1}) \end{aligned} \quad (4.1)$$

, where x_j is a word occurring in side information document, P_{LM} is an N -gram LM estimated using Maximum Likelihood:

$$P_{LM}(x_j|x_{j-N}^{j-1}) = \frac{\text{count}(x_{j-N-1} \dots x_j)}{\text{count}(x_{j-N-1} \dots x_{j-1})} \quad (4.2)$$

and λ_i is defined as

$$\lambda_i = \theta \lambda_{i-1} \text{ where } \theta \in \mathbb{R}, \lambda_i > 0 \text{ and } \sum_i \lambda_i = 1 \quad (4.3)$$

where θ is a hyper-parameter that we tune using the dev dataset during the experiments.

Lastly, we combined the probability distribution of the decoder, P_{s2s} with that provided by the side information, P_{side} , as follows:

$$s(y_j|y_{<j}, z) = \log(a) + \psi * \log\left(\frac{b}{a} + 1\right) \quad (4.4)$$

where ψ is a hyper-parameter determining the influence of the side information on the decoding process, a is $P_{s2s}(y_j|y_{<j}, z)$ and b is $P_{side}(y_j|y_{j-3}^{j-1})$. $s(y_j|y_{<j}, z)$ is used during beam search replacing $P_{s2s}(y_j|y_{<j}, z)$ for all words that occur in the side information. The intuition behind Eq. 4.4 is that we are rewarding word y_j when it appears in similar context in the side information, i.e. the source document being summarized.

We have tried different equations for the combination which can be seen in Figure 4.3.

The different insights for these equations can be seen in Table 4.3.

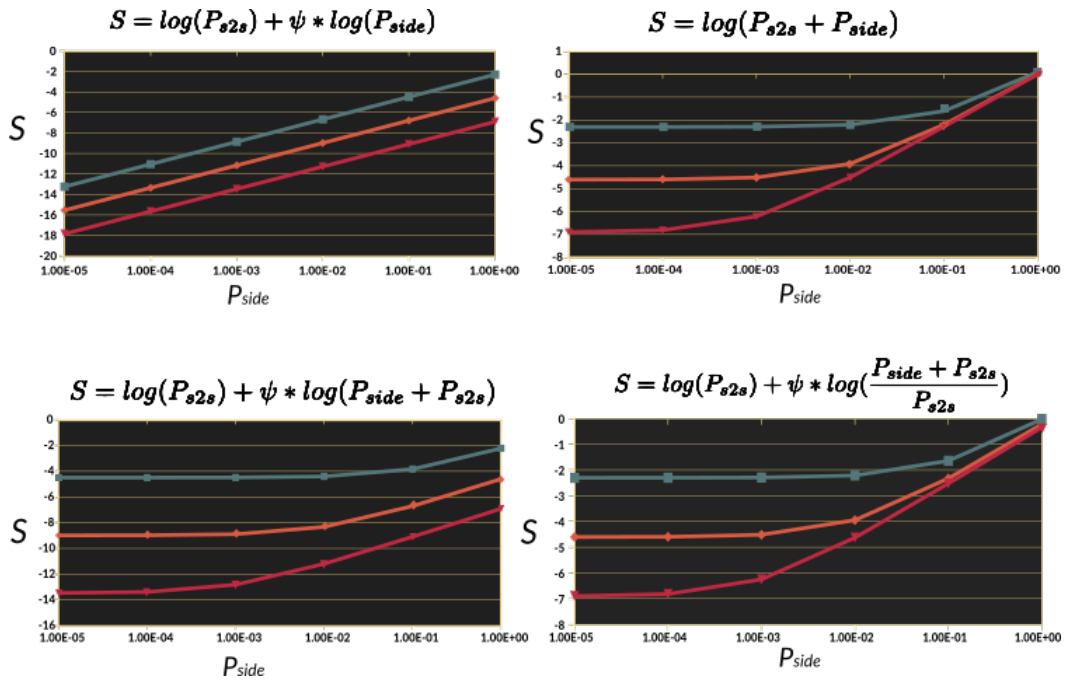


FIGURE 4.3: Effect of different equations for combining the probability distribution of the decoder with the side information.

Combination Equation of s	Insights
$\log(a) + \psi * \log(b)$	A very low side probability could bring down the final score due to logarithm's nature.
$\log(a) + \log(b)$	Only a high side probability can affect the final score and it is difficult to adjust each probability's contribution to the final score.
$\log(a) + \psi * (\log(b) + \log(a))$	Can adjust each probability's contribution using a hyper-parameter. However, the side probability is not very sensitive toward the final score (a large hyper-parameter is needed to improve the sensitivity).
$\log(a) + \psi * \log(\frac{b}{a} + 1)$	Scaling the result using seq2seq probability increases the side probability sensitivity. N-Grams which occur in source document are highly favoured when their probabilities are high.

TABLE 4.3: Insights of different equations.

4.3 Experiments

We conduct experiments in order to answer the following questions about our proposed approach:

1. Is our baseline model comparable with the state-of-the-art AMR-to-text approaches?
2. Does the guidance from the source document improve the result of AMR-to-Text in the context of summarization?
3. Does the improvement in AMR-to-Text hold when we use the generator for abstractive summarization using AMR?

We answer each of these in the following paragraphs.

Model	BLEU
Our model (unguided NLG)	21.1
NeuralAMR (Konstas et al., 2017)	22.0
TSP (Song et al., 2016)	22.4
TreeToStr (Flanigan et al., 2016)	23.0

TABLE 4.4: Results for AMR-to-text

AMR-to-Text baseline comparison We compare our baseline model against previous works in AMR-to-text using the data from the SemEval-2016 Task 8 (May, 2016, LDC2015E86). Table 4.4 reports BLEU scores comparing our model against previous works. Here, we see that our model achieves a BLEU score comparable with the state-of-the-art, and thus we argue that it is sufficient to be used in our subsequent experiments with guidance.

Guided NLG for AMR-to-Text In this experiment we apply our guided NLG mechanism to our baseline seq2seq model. To isolate the effects of guidance we skip the actual summarization process and proceed to directly generate the summary text from the gold standard summary AMR graphs from the Proxy Report section. To determine the model’s hyperparameters, we perform a grid search using the validation dataset, where we found the best combination of ψ , θ and k are 0.95, 2.5 and 15 respectively. We have two different settings for this experiment: oracle and non-oracle settings. In the oracle setting, we directly used the gold standard summary text as the guidance for our model. The intuition is that in this setting, our model knows precisely which words should appear in the summary text, thus providing an upper bound for the performance of our guided NLG approach. We also compared them against the baseline (unguided) model. Table

4.5 reports performance for all models. The difference between the guided and the unguided model is 16.2 points in BLEU and 9.9 points in ROUGE-2, while there is room for improvement as evidenced by the difference between the oracle and non-oracle result.

Model	BLEU	F_1 ROUGE		
		R-1	R-2	R-L
Guided NLG (Oracle)	61.3	79.4	63.7	76.4
Guided NLG	45.8	70.7	49.5	64.9
Unguided NLG	29.6	68.6	39.6	61.3

TABLE 4.5: BLEU and ROUGE results for guided and unguided models using test dataset.

Guided NLG for full summarization In this experiment we combine our guided NLG model with Liu et al. (2015)’s work in order to generate fluent texts from their summary AMR graphs using the hyper-parameters tuned in the previous paragraph. Liu et al. (2015) used parses from both the manual annotation of the Proxy dataset as well as those obtained using the JAMR parser (Flanigan et al., 2014). Instead of JAMR we used the RIGA parser Barzdins and Gosko (2016) which achieved the highest accuracy in the SemEval 2016 Task 8 (May, 2016). We compared our result against Liu et al. (2015)’s bag of words¹, the unguided AMR-to-text model, and a seq2seq summarization model (OpenNMT BRNN)²³ which summarizes directly from the source document to

¹We were able to obtain comparable AMR summarization subgraph prediction to their reported results using their published software but not to match their bag-of-words generation results.

²We use the OpenNMT-PyTorch implementation <https://github.com/OpenNMT/OpenNMT-py> and a pre-trained model downloaded from <http://opennmt.net/OpenNMT-py/Summarization.html> which has higher result than See et al. (2017)’s summarizer.

³The pre-trained model generates multiple sentences summary, but we use only the first sentence summary for evaluation in accordance with the AMR dataset.

summary sentence without using AMR as an interlingua and is trained on CNN/DM corpus (Hermann et al., 2015) using the same settings as See et al. (2017).

AMR parses	NLG Model	F_1 ROUGE		
		R-1	R-2	R-L
Gold	Guided	40.4	20.3	31.4
	Unguided	38.9	12.9	27.0
	Liu et al. (2015)	39.6	6.2	22.1
RIGA	Guided	42.3	21.2	33.6
	Unguided	37.8	10.7	26.9
	Liu et al. (2015)	40.9	5.5	21.4
Directly from Text	OpenNMT BRNN 2 layer, emb 256, hidden 1024	36.1	19.2	31.1

TABLE 4.6: The F_1 ROUGE scores for guided, unguided, Liu et al. (2015) (BoW) results in Gold and RIGA parses, and seq2seq summarization. All models are run using test dataset.

In Table 4.6, we can see that our approach results in improvements over both the unguided AMR-to-text and the standard seq2seq summarization. We run ANOVA analysis for significant testing for the R-2 result between the guided and the unguided for the RIGA parses and found out that the difference is statistically significant where p-value < 0.001 .

One interesting note is that using the RIGA parses results in higher ROUGE scores than the gold parses for the guided model in our experiment. This phenomenon was also observed in Liu et al. (2015)’s experiment where the summary graphs extracted from automatic parses had higher accuracy than those extracted from manual parses. We hypothesize this can be attributed to how the AMR dataset is annotated as there might be discrepancies in different annotator’s choices of AMR concepts and relations for sentences with similar wording. In contrast, the AMR parsers introduce errors, but they are consistent in their choices of AMR concepts and relations. The discrepancies in the

manual annotation could have impacted the performance of the AMR summarizer that we use more negatively than the noise introduced due to the AMR parsing errors.

NLG Model	Generated Summary
Gold	on 8 august 2008 russia conducted airstrikes on georgian targets .
Guided	on 8 august 2008 russia conducted airstrikes on georgian separatist targets .
Unguided	on 8 august 2008 russia conducted a softening of the georgia 's separatist target .
Seq2seq	the russian laboratory complex is a 90 - building campus and served as the location for russia 's secret biological weapons program in the soviet era of a moscow regional depository threaten moscow .

TABLE 4.7: Result summaries of guided, unguided and seq2seq models compared with gold summary.

In Table 4.7, we show sample summaries from the different models, where we can see that our guided model improves the unguided model by correcting a wrong word (*a softening*) into a correct one (*airstrikes*) and introducing a better-suited word from the source document (*georgian* instead of *georgia 's*).

NLG Model	Fluency
Guided	2.66
Unguided	2.16

TABLE 4.8: Fluency scores on test dataset.

We also evaluated manually by asking human evaluators to judge sentences' fluency (grammatical and naturalness) on a scale of 1 (worst) to 6 (best) for the guided and

unguided model (see Table 4.8). While the manual evaluation shows improvement over the unguided model, on the other hand, grammatical mistakes and redundant repetition in the generated text are still major problems (see Table 4.9) in our AMR generation.

Guided NLG Model	Problems
the soldiers were injured when a attempt to defuse the bombs .	grammatical mistake
on 20 october 2002 the state - run radio nepal reported on 20 october 2002 that at the evening - run radio nepal reported on 20 october 2002 that the guerrillas were killed and killed .	redundant repetition

TABLE 4.9: Problems in guided model’s summaries.

4.4 Conclusion and Future Works

In this chapter, we show a guided NLG approach that substantially improves the output of an AMR-based summarization. Our approach used Liu et al. (2015)’s AMR summarization system as the baseline. We improved the baseline by using an AMR-to-Text generator (Konstas et al., 2017) to generate a fluent language from Liu et al. (2015)’s output. However the seq2seq approach is prone to hallucination. To solve it we used a guiding mechanism where we used a language model retrieved from the pruned source

document as the side information to guide the language generation probability distribution. The source document is pruned to k most similar sentence graph-to-graph similarity with the predicted AMR summary from Liu et al. (2015). We have shown that this approach improved the ROUGE-2 score 2 points over the unguided one. We also have shown that our complete system that used gold parses and RIGA (Barzdins and Gosko, 2016) parses improved 7.4 and 10.5 ROUGE-2 score over See et al. (2017)’s approach. With these results, we revisited our research questions as follows.

How to better leverage structural information to address the hallucination and disfluency problems by an abstractive summarization system? The baseline abstractive summarization systems of See et al. (2017) and Liu et al. (2015) have problems listed in the research question, we addressed these problems by leveraging structural information obtained through the source document as the side information. The obtained structural information we then use it to guide the language generation probability distribution. Using automatic and manual evaluation, we showed that our approach reduces hallucination and disfluency.

Can we incorporate structural information into the guiding mechanism without having to re-trained the model every time we change the information source? Our approach didn’t need to be re-trained and we can use a different source document as the guidance during inference time.

For future works, we would like to extend our work to other datasets. At the time of the experiment, the AMR training dataset that we can use is still very small and the AMR parser isn't very good yet. However, in the future, if the AMR parser has matured we think this approach is very feasible for a larger dataset.

Chapter 5

Guided Key-phrase Extraction for More Informative Summarization

In this chapter we propose a guided key-phrase extraction for more informative summarization. The overview diagram for our approach can be seen in Figure 5.1.

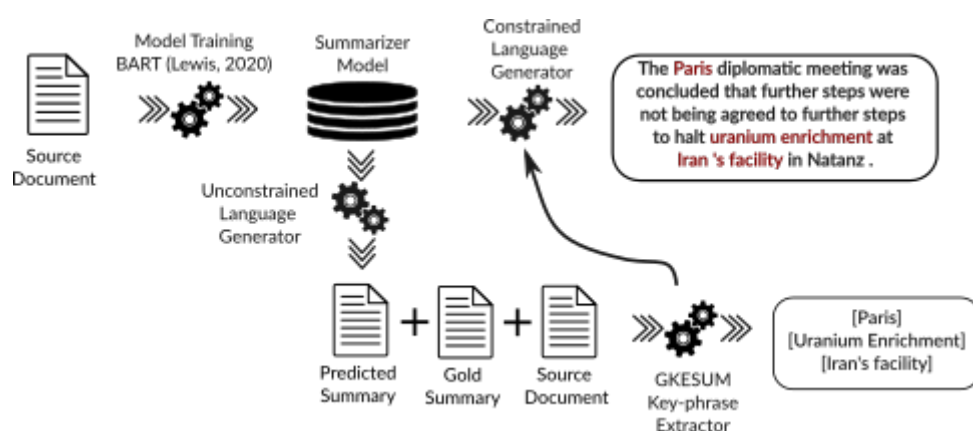


FIGURE 5.1: An overview diagram of the guided key-phrase extraction for more informative summarization.

5.1 Introduction

In recent years, encoder-decoder models (Zhang et al., 2020; Lewis et al., 2020) have made headway on various summarization datasets achieving state-of-the-art results. Despite this, an abstractive summarization system that uses these models still suffers from hallucination. To this end, several researchers (Li et al., 2018a, 2020; Dou et al., 2020) investigated the guidance mechanism to improve existing summarization models. The guidance mechanism uses a guidance signal to control what tokens are to be generated. A guidance signal can be defined as different types of signals that are fed into the model in addition to the source document where a commonly used one is structural information from the source document.

We draw inspiration from our work of manual evaluation (Hardy et al., 2019, Chapter 3) where we showed that highlighted contents can aid human evaluations which leads to a lower variation in evaluation result. Here in this chapter, we hypothesized that by automating how a human selects key-phrases, we can then use those key-phrases to improve the decoder’s decision making. This idea is also inspired by our work in guiding natural language generation using AMR in Hardy et al. (2019, Chapter 4). Our objectives here are:

1. To extract key-phrases that can be used to improve an abstractive summarization system
2. To guide an abstractive summarization system using extracted key-phrases

To date, there are several similar types of research (Li et al., 2018a, 2020; Dou et al., 2020) with ours that used key-phrases for improving the abstractive summarization system¹. However, they were using a joint-training approach for the guiding mechanism, in other words, the model needs to be re-trained if a different guidance signal is used which is costly. We propose a solution that works without re-training and therefore are more flexible with regards to the guidance signal source and also computationally cheaper called Guided Keyphrase Extractor for SUMmarization (GKESUM). We achieve this by training a key-phrase extractor model that learns to extract key-phrases from the source document. The key-phrase extractor is a classification model that is much lighter and cheaper to train than the abstractive model.

To improve the performance of the key-phrase extractor model we included the summarizer model prediction as part of the signal in GKESUM. Extracted key-phrases were then used as constraints for the summarizer model during inference. We used lexically constrained beam search (Post and Vilar, 2018) for enforcing key-phrases into the language generation of the abstractive model. Post and Vilar (2018)'s algorithm enforces that key-phrases with the highest ranking would show up first in the prediction. We hypothesize that this order enforcing could guide the sequence predictions to produce a sequence that is more similar to the reference summary. However, we obtained mixed results and this chapter serves to provide our investigation and analysis to gain insights for future work.

¹We discussed their work in Chapter 2

5.2 Methodology

Our work extends the baseline summarizer model by guiding the model’s predictions using key-phrases that are extracted using feedback from the summarizer model’s mistakes.

Overall, our summarizer system has three steps:

1. Training the summarizer model for which we used BART [Lewis et al. \(2020\)](#).
2. Combining the summarizer’s predictions with the gold summary as part of the input along with the source document for key-phrase extraction training.
3. Using the extracted key-phrase to constrain the summarizer model prediction.

5.2.1 Summarizer model training

An abstractive summarization model, M , takes a document $D = \{w_1, \dots, w_n\}$ as input and produces a summary $S = \{w_1, \dots, w_m\}$. Our approach can be applied on any summarizer model that is based on a seq2seq architecture and uses beam search for decoding. We used the state-of-the-art model, BART ([Lewis et al., 2020](#)) as the baseline summarizer system. We fine-tune BART model on the training dataset. BART predictions however still contain mistakes that are caused by hallucination ([Kryscinski et al., 2020](#); [Zhao et al., 2020](#)). We want to reduce these mistakes therefore we include these mistakes as part of our training process for our keyphrase extractor. To achieve it, we generate summaries from the training dataset for the next step.

5.2.2 GKESUM extraction training

GKESUM is a keyphrase extractor that is based upon Sun et al. (2020)'s BERTKPE. Given a document $D = \{w_1, \dots, w_n\}$ as input where each token has a relevance score, $R = \{r_1, \dots, r_n\}$, GKESUM extracts keyphrases $P = \{p_1, \dots, p_m\}$. Similar to BERTKPE, GKESUM comprises of *chunking* and *ranking networks*, however GKESUM applies a different ranking formulation that takes into account the mistakes of BART model.

Preprocessing GKESUM takes the source document, gold summary and its respective predicted summary generated by BART as inputs. We first extract key-phrases from each gold and predicted summaries using Spacy². We define key-phrases extraction criteria as:

1. Noun chunks
2. Tokens with NOUN, PROPN, VERB, and NUM part-of-speech tags
3. Named entities

Tokens that are not found in the source document were filtered out from those extracted key-phrases. We then use gold and predicted summaries key-phrases to calculate relevance scores. The relevance score, $R = \{r_1, \dots, r_n\}$, is a score that denotes how relevant each key-phrase to the downstream summarizer model in order to achieve the highest ROUGE scores. We used a simple formula where we assign the label '0' when a key-phrase is not

²<https://spacy.io>

found in the gold or predicted summary key-phrases, ‘1’ when it is found in the gold and predicted summary key-phrases, and ‘2’ when it is found in the gold summary key-phrases but not in the predicted summary key-phrases. These ‘0’, ‘1’, and ‘2’ labels denote the ranking of a key-phrase. The idea is that we want key-phrases that are found in gold but couldn’t be satisfied by the summarizer i.e. not found in predicted summary key-phrases, to be prioritized over key-phrases that are found in gold and satisfied by the summarizer during the extraction process.

Token and NGram Embedding Consider the document D , GKESUM uses the BERT pre-trained model to retrieve the encoding $\mathbf{H} = \{\mathbf{h}_1, \dots, \mathbf{h}_n\}$ and then applies CNN to capture the final embedding for each k -gram by sliding a k -sized window over the text. The representation of the i -th k -gram is:

$$\mathbf{g}_i^k = \text{CNN}^k\{\mathbf{h}_i, \dots, \mathbf{h}_{i+k-1}\} \quad (5.1)$$

Chunking Network The key-phrase probability of n-gram c_i^k is calculated by:

$$P(c_i^k = y_i^k) = \text{softmax}(\text{Linear}(\mathbf{g}_i^k)) \quad (5.2)$$

Ranking Network To obtain the ranking, we first apply a linear layer similar to BERTKPE to predict the relevance score of each n-gram representation:

$$\Phi(c_i^k) = \text{Linear}(\mathbf{g}_i^k) \quad (5.3)$$

For n-grams that occur several times in the document, we take the one with the maximum score.

Chunking Loss Function The chunking loss function is the cross-entropy loss which is as follows:

$$\mathcal{L}(c_i^k) = \text{CrossEntropy}(P(c_i^k = y_i^k)) \quad (5.4)$$

Ranking Loss Function Given the scores, Φ , from the ranking network, we want to maximize the ranking using relevance scores, R . This can be done by maximizing the Normalized Discounted Cumulative Gain (Järvelin and Kekäläinen, 2002, NDCG) which states:

1. Highly relevant items are more useful when appearing earlier in the ranking
2. Highly relevant items are more useful than marginally relevant items, which are in turn more useful than irrelevant items.

using the following equation:

$$\text{DCG}_p = \sum_{i=1}^p \frac{2^{r_i} - 1}{\log_2(i + 1)} \quad (5.5)$$

where p is the position in the ranking and r_i is the gold relevance score. We use the LambdaLoss (Wang et al., 2018) that directly maximizes the NDCG as follows:

$$\mathcal{L}(\Phi) = \sum_{\mathbf{c}, \mathbf{r} \in D} \sum_{r_i > r_j} \Delta \text{NDCG}(i, j) \log_2(1 + e^{-\sigma(c_i - c_j)}) \quad (5.6)$$

where \mathbf{c} is the set of all n -grams and \mathbf{r} is the set of the relevance scores of all n -grams. NDCG is a length normalized DCG as follows.

$$\text{NDCG} = \frac{1}{\max \text{DCG}} \sum_{i=1}^n \frac{2^{r_i} - 1}{\log_2(1 + i)} \quad (5.7)$$

Since the set of $|\mathbf{c}|$ is very large due to the large number of the possible key-phrase n -grams, we therefore measure the loss using the top-100 ranking key-phrases.

Combined Loss For total loss, we use a weighted addition as follows:

$$\mathcal{L} = \alpha \mathcal{L}_{\text{chunk}} + \mathcal{L}_{\text{rank}} \quad (5.8)$$

where α is a hyperparameter denoting the significance of the loss function. We set α to 100 since the ranking loss is usually much large.

5.2.3 Summarizer model inference

To guide the summarizer during decoding, we use a lexical constrained beam search (Post and Vilar, 2018) during inference stage. For the constraints, we used the prediction of GKESUM.

5.3 Experimental Setup and Result

5.3.1 Data

We used three datasets that cover three different domains: XSum (Narayan et al., 2018c, news), Reddit-TIFU(Kim et al., 2018, electronic board), and Wikihow(Koupaee and Wang, 2018, instructional). We chose these datasets because of their highly abstractive nature and to demonstrate how our approach works in different domains.

5.3.2 Model

We used BART(Lewis et al., 2020) as the baseline to showcase our approach. To further see how our approach works in different settings, we also experimented with two variants of BART: chronological and distilled models. The chronological model is a BART model that is trained on XSUM dataset but only on articles that are published prior to the year of 2017, while the validation and test dataset is taken from after 2017. The BART

distilled model will be used to demonstrate how our approach works on a weaker version of the model.

5.3.3 Evaluation for the Key-phrase Extractor Model

We compare our GKE-SUM with [Sun et al. \(2020\)](#)'s BERTKPE in terms of precision, recall and F1 scores that can be seen in [Table 5.1](#). We benchmark it using the REDDIT-TIFU dataset. The BART-GKESUM is trained using the predictions of BART. We can see that BART-GKESUM has better performance than BERTKPE.

Model	P@1	R@1	F@1	P@3	R@3	F@3	P@5	R@5	F@5
BART-GKESUM	0.608	0.062	0.11	0.501	0.149	0.219	0.431	0.21	0.266
BERTKPE	0.566	0.057	0.101	0.465	0.137	0.201	0.399	0.192	0.245

TABLE 5.1: GKE-SUM and BERTKPE performance using Recall, Precision and F1 scores on one, three and five key-phrases size

5.3.4 Automatic Evaluation for the Summarizer Model

For automatic evaluation, we used ROUGE ([Lin, 2004](#)) to measure the similarity between reference and system summaries. [Table 5.2](#) shows the result where GKESUM is used for BART model. We show that our approach improves the baseline on Wikihow and Reddit datasets. However, our approach doesn't work on XSUM. We think this is due to the nature of XSUM dataset where the summary is the first line of the article therefore it contain new information that isn't found in the article which is not suitable to our

approach that constrains the summary to contain more information from the source document.

In chronological and distilled settings the numbers are, however, comparable between baseline and constrained. We think this is because the weakened models made more mistakes in their prediction therefore our approach show comparable result.

We also run oracle-constrained BART-GKESUM for XSUM dataset (see Table 5.4) where oracle-constraints are constraints that are missing in predicted summaries by BART. It can be seen that the GKE-SUM can indeed improve the prediction if given the right key-phrases.

	BART	BART-GKESUM
XSum	45.45/22.30/37.18 36.75	43.68/21.25/35.49
Wikihow	43.42/19.11/34.54	43.83/19.34/34.76
Reddit TIFU	28.88/10.40/23.16	30.72/10.24/23.95

TABLE 5.2: Results on three abstractive summarization datasets.

Table 5.3 shows BART in different settings: chronological and distilled.

	BART	BART-GKESUM
XSUM Chronological	42.59/19.74/34.52	42.68/19.56/34.32
XSUM Distilled	40.14/18.00/33.33	40.52/17.52/33.06

TABLE 5.3: Results on XSUM dataset with different settings. The chronological is fine-tuned on the year prior to 2017 and tested on after 2017. The distilled is XSUM fine-tuned using the distilled version of BART

Model	BART-GKESUM
Oracle	54.57/25.39/38.39

TABLE 5.4: Oracle constraints results of BART-GKESUM on XSUM dataset.

5.3.5 Human Evaluation

We chose REDDIT-TIFU dataset for our human evaluation since it obtain the best result from Table 5.2. We used HIGHRES (Hardy et al., 2019, Chapter 3) for evaluating system summaries, however since we don't have any highlight annotation for REDDIT-TIFU dataset we used the reference-based evaluation. Using HighRES, we measured *Precision* and *Recall* of information in the baseline and constrained system summaries. Recall denotes “all important information is present in the summary” while Precision denotes “only important information is present in the summary”. We also measured linguistic qualities: clarity and fluency of summaries. In clarity evaluation, each evaluator is asked whether the summary is easy to be understood while in fluency evaluation, each evaluator is asked whether the summary sounds natural and has no grammatical problems. For each system, we evaluate 50 randomly selected summaries with each summary evaluated by three evaluators. All of our evaluations were done using the Amazon Mechanical Turk platform. The results can be seen in Table 5.5.

In terms of linguistic quality, BART-GKESUM shows improvement in clarity but deterioration in fluency. Meanwhile the same mixed results are shown in informativeness quality where BART-GKESUM improves over precision but deteriorates in the recall. We think the improvement in the precision score could be attributed to more informative key-phrases occurrence in the summary meanwhile the deterioration of the recall score could be due to more hallucinations occurrence which also explain the drop in fluency.

	BART	BART-GKESUM
Clarity	45.43	46.48
Fluency	56.5	52.59
Precision	45.43	46.48
Recall	56.5	52.59

TABLE 5.5: Human evaluation results of BART and BART-GKESUM’s generated summaries on Reddit.

5.3.6 Qualitative Analysis

NLG Model	Generated Summary	Constraints
Gold	aunt has old car. she said i can drive it. the key my aunt said was stuck was not stuck, but now won’t go back in the ignition. and then i saw midget porn.	N/A
BART	i tried to start my aunt’s mg midget with a volvo key and now i don’t know if it’s a common issue or not.	N/A
BART-GKESUM	i tried to start my aunt’s mg midget with a volvo key , and now i’m freaking out about what to do. car won’t start, but the key is stuck in the ignition .	key, car, aunt, ignition, i, it, mg, stuck, tried

TABLE 5.6: A positive sample shown in summaries of BART, BART-GKESUM and gold. Constraints provide missing information and makes BART-GKESUM’s summary more informative.

We performed qualitative analysis on summaries produced by BART and BART-GKESUM to seek more insights into the performance of BART-GKESUM. Table 5.6 and Table 5.7 show a positive and negative sample with regards to BART-GKESUM’s performance. In the positive sample, we can see that BART-GKESUM was able to produce key-phrases that relate to missing information in the baseline BART and therefore improved the informativeness of the summary. However, in the negative sample

NLG Model	Generated Summary	Constraints
Gold	i made a video for a french proyect with too much memes, when my classmates saw it they were really disturbed..	N/A
BART	made a video as a proyect, sent it to teacher, class watched it, made akward memes, teacher only says "well... everyone can have artistic liberty"	N/A
BART-GKESUM	made a graduation video , uploaded it to youtube, class watched it , memes started appearing, everyone was akward, teacher only says "well... everyone can have artistic liberty"	video, class, it, memes

TABLE 5.7: A negative sample shown in summaries of BART, BART-GKESUM and gold. Adding constraints that have already been satisfied by the summarization model causes hallucination in BART-GKESUM’s summary

BART-GKESUM produce key-phrases that are already satisfied by the summarizer and negatively affects the language generation process causing several hallucinations to occur in the summary. This negative effect occurs because constrained beam-search ([Post and Vilar, 2018](#)) prioritizes satisfying constraints over minimizing perplexity, therefore, generating sentences with higher perplexity. This works well only if constraints added more information.

5.4 Conclusion and Future Work

In this chapter, we showed BART-GKESUM which is a guided key-phrase extractor for more informative summarization. We also showed that BART-GKESUM works better as a key-phrase extractor compared to the baseline BERTKPE (Sun et al., 2020). With regards to the quality of abstractive summarization, BART-GKESUM gave mixed results. Although it has better ROUGE scores in REDDIT-TIFU and WIKIHOW, it failed to improve results in the XSUM dataset. It also gave mixed results in the human evaluation. Despite these, we have shown on the oracle dataset, BART-GKESUM could improve the result substantially if it is able to identify unsatisfied key-phrases most of the time. We also have analyzed both the positive and negative sample from BART-GKESUM’s output.

With regards to the research question, we see that this experiment answers partially the second research question which is **“Can we incorporate structural information into the guiding mechanism without having to re-trained the model every time we change the in-formation source?”**. Despite our mixed results, we showed that our oracle results have shown a promising result for future work. We think that improving the key-phrase extractor’s accuracy and ranking would increase the chance of producing a more informative summary.

Chapter 6

Conclusion and Future Work

In this Thesis, we have investigated the topic of guiding abstractive summarization using structural information as well as devising a new manual evaluation approach that is reference bias free. There are three experiments in this Thesis that are done to address our research questions.

In the first experiment (Chapter 3, we introduced the HIGHLIGHT-based Reference-less Evaluation Summarization (HIGHRES) framework for manual evaluation. The proposed framework avoids reference bias and provides absolute instead of ranked evaluation of the systems. We also performed crowd-sourced annotations to obtain a highlighted source document for the purpose of HIGHRESevaluation. This annotated dataset also gives more understanding of how humans select salient information from the source document. Our experiments show that HIGHRES lowers the variability of the judges' content assessment while helping expose the differences between systems. We also showed that by

evaluating clarity we are able to capture a different dimension of summarization quality that is not captured by the commonly used fluency.

In the second experiment (Chapter 4, we have built a guided NLG approach that substantially improves the output of an AMR-based summarization. Our approach uses Liu et al. (2015)'s AMR summarization as the baseline. We improve the baseline by using an AMR-to-Text generator (Konstas et al., 2017) to generate a fluent language from Liu et al. (2015)'s output. However the seq2seq approach has shortcomings, i.e.: hallucinations, grammatical mistakes, irrelevancy and other. To solve it we use a guiding mechanism where we use a language model retrieved from the pruned source document as side information to guide the language generation probability distribution. The source document is pruned to k most similar sentence graph-to-graph similarity with the predicted AMR summary from Liu et al. (2015). We have shown that this approach improved the ROUGE-2 score 2 points over the unguided one. We also have shown that our complete system that used gold parses and RIGA (Barzdins and Gosko, 2016) parses improved 7.4 and 10.5 ROUGE-2 score over See et al. (2017)'s approach.

In the third experiment (Chapter 5, we have built BART-GKESUM which is a guided key-phrase extractor for more informative summarization. We have shown that BART-GKESUM works better as a key-phrase extractor compared to the baseline BERTKPE (Sun et al., 2020). With regards to the improvement of abstractive summarization, BART-GKESUM gives mixed results. Although it has better ROUGE scores in REDDIT-TIFU and WIKIHOW, it fails to improve XSUM dataset. It also gives mixed results in the human evaluation. Despite these, we have shown on an oracle dataset,

BART-GKESUM could improve the result substantially if it is able to identify unsatisfied key-phrases most of the time. We also have analyzed both the positive and negative sample from BART-GKESUM's output.

6.1 Summary of Findings

We revisited each research question that motivated our work and summarise contributions for each of our experiments.

Can we devise a new manual evaluation that is not affected by the reference bias problem? We answered this question in the first experiment as well where we developed a new framework that addresses the reference bias problem which consists of a web application and annotated datasets where annotators can use it to annotate a new highlighted document or use the existing dataset to evaluate systems' summaries.

How to better leverage structural information to address the hallucination, grammatical mistakes, irrelevancy and disfluency problems by an abstractive summarization system? We answered this question in the second experiment where the baseline abstractive summarization systems of [See et al. \(2017\)](#) and [Liu et al. \(2015\)](#) have problems listed in the first research question. We addressed these problems by leveraging structural information obtained through the source document. The obtained

structural information we then use to guide the language generation probability distribution. Our automatic and manual evaluation have shown that our approach reduces hallucination, grammatical mistakes, irrelevancy and disfluency.

Can we incorporate structural information into the guiding mechanism without having to re-train the model every time we change the information source? We answered this question in the second and third experiments. In those two experiments, our approach didn't need to be re-trained.

6.2 Future Work

For future works, we would like to:

1. Expand the HighRES approach to more summarization datasets.
2. Extend our work in Chapter 4 to other datasets. At the time of the experiment, the AMR training dataset that we can use is still very small and the AMR parser isn't very good yet. However, in the future, if the AMR parser has matured we think this approach is very feasible for a larger dataset.
3. Improve our approach in Chapter 5 by improving the keyphrase extractor's accuracy and ranking so that it has more chance to produce an informative summary.

Appendix A

DUC 2003 Summary Quality

Questions

12 questions for assessing the summary quality, which are as follows.

1. About how many gross capitalization errors are there?
2. About how many sentences have incorrect word order?
3. About how many times does the subject fail to agree in number with the verb?
4. About how many of the sentences are missing important components (e.g. the subject, main verb, direct object, modifier) – causing the sentence to be ungrammatical, unclear, or misleading?
5. About many times are unrelated fragments joined into one sentence?

6. About how many times are articles (a, an, the) missing or used incorrectly?
7. About how many pronouns are there whose antecedents are incorrect, unclear, missing, or come only later?
8. For about how many nouns is it impossible to determine clearly who or what they refer to?
9. About how times should a noun or noun phrase have been replaced with a pronoun?
10. About how many dangling conjunctions are there ("and", "however"...)?
11. About many instances of unnecessarily repeated information are there?
12. About how many sentences strike you as being in the wrong place because they indicate a strange time sequence, suggest a wrong cause-effect relationship, or just don't fit in topically with neighboring sentences?

Appendix B

DUC2004 Summary Quality

Question

1 Does the summary build from sentence to sentence to a coherent

body of information about the topic?

- A. Very coherently
- B. Somewhat coherently
- C. Neutral as to coherence
- D. Not so coherently
- E. Incoherent

2 If you were editing the summary to make it more concise and

to the point, how much useless, confusing or repetitive text would

you remove from the existing summary?

- A. None
- B. A little
- C. Some
- D. A lot
- E. Most of the text

3 To what degree does the summary say the same thing over again?

- A. None; the summary has no repeated information
- B. Minor repetitions
- C. Some repetition
- D. More than half of the text is repetitive
- E. Quite a lot; most sentences are repetitive

4 How much trouble did you have identifying the referents of noun

phrases in this summary? Are there nouns, pronouns or personal names that are not well-specified? For example, a person is mentioned and it is not clear what his role in the story is, or any other entity that is referenced but its identity and relation with the story remains unclear

- A. No problems; it is clear who/what is being referred to throughout.
- B. Slight problems, mostly cosmetic/stylistic

- C. Somewhat problematic; some minor events/things/people/places are unclear, or a very few major ones, but overall the who and what are clear.
- D. Rather problematic; enough events/things/people/places are unclear that parts of the summary are hard to understand
- E. Severe problems; main events, characters or places are not well-specified and/or it's difficult to say how they relate to the topic

5 To what degree do you think the entities (person/thing/event/place/...) were re-mentioned in an overly explicit way, so that readability was impaired? For example, a pronoun could have been used instead of a lengthy description, or a shorter description would have been more appropriate?

- A. None: references to entities were acceptably explicit
- B. A little: once or twice, an entity was over-described
- C. Somewhat: to a noticeable but not annoying degree, some entities were over-described
- D. Rather problematic: to a degree that became distracting, entities were over-described
- E. A lot: reintroduction of characters and entities made reading difficult/caused comprehension problems

6 Are there any obviously ungrammatical sentences, e.g., missing components, unrelated fragments or any other grammar-related problem that makes the text difficult to read.

- A. No noticeable grammatical problems
- B. Minor grammar problems
- C. Some problems, but overall acceptable
- D. A fair amount of grammatical errors
- E. Too many problems, the summary is impossible to read

7 Are there any datelines, system-internal formatting or capitalization errors that can make the reading of the summary difficult?

- A. No noticeable formatting problems
- B. Minor formatting problems
- C. Some, but they do not create any major difficulties
- D. A fair amount of formatting problems
- E. Many, to an extent that reading is difficult

Appendix C

DUC2005 Summary Quality

Question

C.1 Readability Task

The linguistic quality questions are targeted to assess how readable and fluent the summaries are, and they measure qualities of the summary that DO NOT involve comparison with a model summary or DUC topic. The information content and responsiveness of the summary are measured separately in another part of the evaluation.

All linguistic quality questions require a certain readability property to be assessed on a five-point scale from "A" to "E", where

"A" indicates that the summary is good with the respect to the quality under question, "E" indicates that the summary is bad with respect to the quality stated in the question, and "B" to "D" show the gradation in between. For each question, please try to assess the quality of the summary only with respect to the property that is described in the question.

1 Grammaticality

The summary should have no datelines, system-internal formatting, capitalization errors or obviously ungrammatical sentences (e.g., fragments, missing components) that make the text difficult to read.

- A. Very Good
- B. Good
- C. Barely Acceptable
- D. Poor
- E. Very Poor

2 Non-redundancy

There should be no unnecessary repetition in the summary.

Unnecessary repetition might take the form of whole sentences that are repeated, or repeated facts, or the repeated use of a noun or noun phrase (e.g., "Bill Clinton") when a pronoun ("he") would suffice.

- A. Very Good
- B. Good
- C. Barely Acceptable
- D. Poor
- E. Very Poor

3 Referential clarity

It should be easy to identify who or what the pronouns and noun phrases in the summary are referring to. If a person or other entity is mentioned, it should be clear what their role in the story is. So, a reference would be unclear if an entity is referenced but its identity or relation to the story remains unclear.

- A. Very Good
- B. Good
- C. Barely Acceptable
- D. Poor
- E. Very Poor

4 Focus

The summary should have a focus; sentences should only contain information that is related to the rest of the summary.

- A. Very Good
- B. Good
- C. Barely Acceptable
- D. Poor
- E. Very Poor

5 Structure and Coherence

The summary should be well-structured and well-organized. The summary should not just be a heap of related information, but should build from sentence to sentence to a coherent body of information about a topic.

- A. Very Good
- B. Good
- C. Barely Acceptable
- D. Poor
- E. Very Poor

C.2 Responsiveness Task

You have been given a topic statement and simple user profile, along with a file containing a number of summaries that contribute toward satisfying the information need expressed in the topic. Some of the summaries may be more responsive to the topic than others. Your task is to help us understand how relatively well each summary responds to the topic.

Read the topic statement and all the associated summaries. Then grade each summary according to how responsive it is to the topic

RELATIVE TO THE OTHERS:

1 2 3 4 5 (1 = worst, 5 = best)

Responsiveness should be measured primarily in terms of the amount of information in the summary that actually helps to satisfy the information need expressed in the topic statement, at the level of granularity requested in the user profile. The linguistic quality of the summary should play a role in your assessment only insofar as it interferes with the expression of information and reduces the amount of information that is conveyed.

Bibliography

Lasha Abzianidze, Johan Bos, and Stephan Oepen. Drs at mrp 2020: Dressing up discourse representation structures as graphs. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 23–32, 2020.

Mb Almeida and Aft Martins. Fast and robust compressive summarization with dual decomposition and multi-task learning. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 196–206, 2013. URL <http://www.newdesign.aclweb.org/anthology/P/P13/P13-1020.pdf>.

Reinald Kim Amplayo, Seonjae Lim, and Seung-Won Hwang. Entity Commonsense Representation for Neural Abstractive Summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 697–707, 2018. doi: 10.18653/v1/N18-1064. URL <https://github.com/idio/wiki2vec><http://arxiv.org/abs/1806.05504>.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer Normalization. In *NIPS 2016 Deep Learning Symposium*, jul 2016. URL <http://arxiv.org/abs/1607.06450>.

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation By Jointly Learning To Align and Translate. *Iclr 2015*, pages 1–15, 2015. ISSN 0147-006X. doi: 10.1146/annurev.neuro.26.041002.131047. URL <http://arxiv.org/abs/1409.0473v3>.
- Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, 2013.
- Guntis Barzdins and Didzis Gosko. RIGA at SemEval-2016 Task 8: Impact of Smatch Extensions and Character-Level Neural Translation on AMR Parsing Accuracy. *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1143–1147, 2016. URL <http://aclweb.org/anthology/S16-1176>.
- Regina Barzilay and Kathleen R. McKeown. Sentence Fusion for Multidocument News Summarization. *Computational Linguistics*, 31(3):297–328, 2005. ISSN 0891-2017. doi: 10.1162/089120105774321091.
- Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A Neural Probabilistic Language Model. *The Journal of Machine Learning Research*, 3:1137–1155, 2003. ISSN 15324435. doi: 10.1162/153244303322533223.
- Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. Jointly Learning to Extract and Compress. *Proc. of ACL*, pages 481–490, 2011. URL <http://www.aclweb.org/anthology/P11-1049>.

Ondřej Bojar, Miloš Ercegovčević, Martin Popel, and Omar Zaidan. A Grain of Salt for the WMT Manual Evaluation. *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 1–11, 2011. URL <http://www.aclweb.org/anthology/W11-2101>.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. Findings of the 2016 Conference on Machine Translation. *Proceedings of the First Conference on Machine Translation*, 2:131–198, 2016. URL <http://www.aclweb.org/anthology/W/W16/W16-2301>.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. Findings of the 2017 Conference on Machine Translation (WMT17). *Wmt-2017*, 2017.

Ziqiang Cao, Furu Wei, Li Dong, Sujian Li, and Ming Zhou. Ranking with recursive neural networks and its application to multi-document summarization. *Aaai*, pages 2153–2159, 2015. ISSN 19909772. doi: 10.1162/153244303322533223.

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. Retrieve, Rerank and Rewrite: Soft Template Based Neural Summarization. *Proceedings of the 56th Annual Meeting of the*

- Association for Computational Linguistics (Volume 1: Long Papers)*, 1:152–161, 2018.
URL <https://aclanthology.info/papers/P18-1015/p18-1015>.
- J Carbonell and J Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336, 1998. ISSN 01635840 (ISSN). doi: 10.1145/290941.291025. URL <papers2://publication/uuid/1FA33AEC-2C9E-4149-B740-02A7C6C24B93>.
- Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. Deep Communicating Agents for Abstractive Summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018. doi: 10.1016/0304-4165(86)90151-0. URL <https://arxiv.org/pdf/1803.10357.pdf><http://arxiv.org/abs/1803.10357>.
- Rajen Chatterjee, Matteo Negri, Marco Turchi, Marcello Federico, Lucia Specia, Frédéric Blain, Fondazione Bruno Kessler, and Italy Trento. Guiding Neural Machine Translation Decoding with External Knowledge. *Research Papers*, 1:157–168, 2017. URL <http://www.aclweb.org/anthology/W17-4716>.
- Danqi Chen and Christopher Manning. A Fast and Accurate Dependency Parser using Neural Networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, 2014. ISBN 9781937284961.

doi: 10.3115/v1/D14-1082. URL <http://www.aclweb.org/anthology/D14-1082><http://aclweb.org/anthology/D14-1082>.

Yen-Chun Chen and Mohit Bansal. Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1:675–686, 2018. URL <https://aclanthology.info/papers/P18-1063/p18-1063>.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014. ISSN 09205691. doi: 10.3115/v1/D14-1179. URL <http://arxiv.org/abs/1406.1078>.

Sumit Chopra, Michael Auli, and Alexander M. Rush. Abstractive Sentence Summarization with Attentive Recurrent Neural Networks. In *Proceedings of the 15th Annual Conference of the NAACL HLT*, pages 93–98, 2016. ISBN 9781941643914.

James Clarke and Mirella Lapata. Global inference for sentence compression an integer linear programming approach. *Journal of Artificial Intelligence Research*, 31:399–429, 2008a. ISSN 10769757. doi: 10.1613/jair.2433.

James Clarke and Mirella Lapata. Global Inference for Sentence Compression : An Integer Linear Programming Approach. *Journal of Artificial Intelligence Research*, 31:399–429, 2008b. URL <http://www.aaai.org/Papers/JAIR/Vol31/JAIR-3112.pdf>.

- James Clarke and Mirella Lapata. Discourse constraints for document compression. *Computational Linguistics*, 36(3):411–441, 2010. ISSN 08912017. doi: 10.1162/coli_a.00004. URL https://www.mitpressjournals.org/doi/pdfplus/10.1162/coli_{-}a_{-}00004.
- Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*., pages 615–621, 2018. URL <https://github.com/acohan/long-summarization>.
- Hoa Trang Dang. Overview of DUC 2005. In *Proceedings of the document understanding conference*, volume 2005, pages 1–12, 2005. ISBN 1932432795. doi: 10.1.1.184.2425. URL <http://www.isi.edu/http://portal.acm.org/citation.cfm?doid=1654679.1654689>.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota, oct 2019. Association for Computational Linguistics. URL <http://arxiv.org/abs/1810.04805>.
- Zi Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. GSum: A General Framework for Guided Neural Abstractive Summarization. In *Association*

for *Computational Linguistics*, 2020.

H P Edmundson. New Methods in Automatic Extracting. *Journal of the ACM (JACM)*, 16(2):264–285, 1969. URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.83.5638&rep=rep1&type=pdf>.

Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990. ISSN 03640213. doi: 10.1016/0364-0213(90)90002-E.

Brian S Everitt. *The Cambridge dictionary of statistics*. Cambridge University Press, 2006.

Katja Filippova and Michael Strube. Sentence fusion via dependency graph compression. *Proceedings of the Conference on Empirical ...*, (October):177–185, 2008. doi: 10.3115/1613715.1613741. URL http://dl.acm.org/citation.cfm?id=1613715.1613741&http://dl.acm.org/ft_gateway.cfm?id=1613741&type=pdf&http://dl.acm.org/citation.cfm?id=1613741.

Charles J Fillmore and Collin Baker. A frames approach to semantic analysis. In *The Oxford handbook of linguistic analysis*. 2010.

Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah a Smith. A Discriminative Graph-Based Parser for the Abstract Meaning Representation. *Acl*, pages 1426–1436, 2014.

Jeffrey Flanigan, Chris Dyer, Noah A Smith, and Jaime Carbonell. Generation from Abstract Meaning Representation using Tree Transducers. In *Proceedings of the 2016*

- Conference of the NAACL*, pages 731–739, 2016. ISBN 9781941643914. URL <http://github.com/jflanigan/jamr>.
- Joseph L Fleiss. Measuring Nominal Scale Agreement Among Many Raters. *Psychological Bulletin*, 76(5):378–382, 1971.
- Marina Fomicheva and Lucia Specia. Reference bias in monolingual machine translation evaluation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 77–82, 2016.
- Satoru Fujishige. *Submodular Functions and Optimization*. Elsevier, 2005. ISBN 9780444520869. doi: 10.1016/S0167-5060(05)80009-3. URL <http://www.sciencedirect.com/science/article/pii/S0167506005800093>.
- Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional Sequence to Sequence Learning. In *International Conference on Machine Learning*, 2017. ISBN 9781510827585. URL <http://arxiv.org/abs/1705.03122>.
- Yoav Goldberg. *Neural Network Methods for Natural Language Processing*. Morgan & Claypool Publisher, synthesis edition, 2017.
- Philip John Gorinski and Mirella Lapata. Movie Script Summarization as Graph-based Scene Extraction. *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1066–1076, 2015. URL <http://en.wikipedia.orghttp://anthology.aclweb.org/N/N15/N15-1113.pdf>{%}0A<http://www.aclweb.org/anthology/N15-1113>.

- David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. English gigaword, 2003a.
- David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34, 2003b.
- Jonas Groschwitz, Meaghan Fowlie, Mark Johnson, Alexander Koller, Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos Santos, Caglar Gulcehre, Bing Xiang, Ani Nenkova, Han Guo, Ramakanth Pasunuru, Mohit Bansal, Vasin Punyakanok, Dan Roth, Wen-Tau Yih, Dav Zimak, Ani Nenkova, R Passonneau, Sebastian Riedel, James Clarke, You Ouyang, Wenjie Li, Sujian Si Li, Qin Lu, Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, Noah a. Smith, Nima Pourdamghani, Kevin Knight, Ulf Hermjakob, Chenliang Li, Weiran Xu, Sujian Si Li, Sheng Gao, Karen Sparck Jones, Julia R Galliers, Eliyahu Kiperwasser, Yoav Goldberg, André F. T. Martins, Noah a. Smith, Yasuhisa Yoshida, Jun Suzuki, Tsutomu Hirao, Masaaki Nagata, Sumit Chopra, Michael Auli, Alexander M. Rush, John Duchi, Elad Hazan, Yoram Singer, David Graff, Junbo Kong, Ke Chen, Kazuaki Maeda, Rajen Chatterjee, Matteo Negri, Marco Turchi, Marcello Federico, Lucia Specia, Frédéric Blain, Fondazione Bruno Kessler, Italy Trento, Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, Luke Zettlemoyer, H P Edmundson, Leon Nelson Flint, Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, Alexander M. Rush, Jordan J Louviere, Terry N Flynn, Anthony Alfred Fred, John Marley, Brian S Everitt, Hoa Trang Dang, Han Guo, Ramakanth Pasunuru, Mohit Bansal, Shashi Narayan, Ronald Cardenas, Nikos Pappas, Yannis Pasouridis, Shay B Cohen, Mirella Lapata, Jiangsheng Yu, Yi Chang, Yen-Chun Chen, Mohit Bansal, Ramakanth Pasunuru, and Mohit Bansal. Soft Layer-Specific

- Multi-Task Summarization with Entailment and Question Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 687–697, Philadelphia., mar 2018. Association for Computational Linguistics. ISBN 9781604234497. doi: 10.18653/v1/N18-2102. URL <http://aclweb.org/anthology/N18-2102><http://aclweb.org/anthology/P18-1063><http://aclweb.org/anthology/N18-1158><http://aclweb.org/anthology/P18-1064><http://arxiv.org/abs/1709.03815><http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.83.5638&rep=rep1&t>.
- Max Grusky, Mor Naaman, and Yoav Artzi. Newsroom: A Dataset of 1.3 Million Summaries with Diverse Extractive Strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719. Association for Computational Linguistics, 2018. URL <http://aclweb.org/anthology/N18-1065>.
- Hardy, Shashi Narayan, and Andreas Vlachos. HighRES: Highlight-based Referenceless Evaluation of Summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, jun 2019. URL <http://arxiv.org/abs/1906.01361>.
- Hardy Hardy and Andreas Vlachos. Guided Neural Language Generation for Abstractive Summarization using Abstract Meaning Representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 768–773. Association for Computational Linguistics, 2018. URL <http://aclweb.org/anthology/>

D18-1086.

Donna Harman and Paul Over. The Effects of Human Variation in DUC Summarization Evaluation. Technical report, 2004. URL www.isi.edu/.

Qiuxiang He, Guoping Huang, Qu Cui, Li Li, and Lemao Liu. Fast and accurate neural machine translation with translation memory. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3170–3180, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.246. URL <https://aclanthology.org/2021.acl-long.246>.

Dan Hendrycks and Kevin Gimpel. Gaussian Error Linear Units (GELUs). Technical report, 2016. URL <http://arxiv.org/abs/1606.08415>.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching Machines to Read and Comprehend. In *Neural Information Processing Systems*, pages 1–14, 2015. ISBN 1045-9227. doi: 10.1109/72.410363. URL <http://arxiv.org/abs/1506.03340>.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL <https://doi.org/10.1162/neco.1997.9.8.1735>.

Chris Hokamp and Qun Liu. Lexically Constrained Decoding for Sequence Generation Using Grid Beam Search. In *Proceedings of the 55th Annual Meeting of the Association*

- for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, 2017. doi: 10.18653/v1/P17-1141. URL <https://doi.org/10.18653/v1/P17-1141>.
- Wan-Ting Hsu, Chieh-Kai Lin, Ming-Ying Lee, Kerui Min, Jing Tang, and Min Sun. A Unified Model for Extractive and Abstractive Summarization using Inconsistency Loss. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1:132–141, 2018. URL <https://aclanthology.info/papers/P18-1013/p18-1013>.
- Aishwarya Jadhav and Vaibhav Rajan. Extractive Summarization with SWAP-NET Sentences and Words from Alternating Pointer Networks. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018.
- Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002. ISSN 10468188. doi: 10.1145/582415.582418.
- Hanqi Jin, Tianming Wang, and Xiaojun Wan. SemSUM: Semantic dependency guided neural abstractive summarization. *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, pages 8026–8033, 2020. ISSN 2159-5399. doi: 10.1609/aaai.v34i05.6312.
- Karen Sparck Jones and Julia R Galliers. *Evaluating natural language processing systems: An analysis and review*. Springer Science & Business Media, 1995.
- Chris Kedzie, Kathleen McKeown, and Hal Daumé III. Content Selection in Deep Learning Models of Summarization. In *Proceedings of the 2018 Conference on Empirical*

- Methods in Natural Language Processing*, pages 1818—1828, Brussels, Belgium, 2018. Association for Computational Linguistics. URL <https://github.com/kedz/nnsun/>.
- Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. Abstractive Summarization of Reddit Posts with Multi-level Memory Networks. In *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, volume 1, pages 2519–2531. Association for Computational Linguistics (ACL), nov 2018. URL <http://arxiv.org/abs/1811.00783>.
- Kevin Knight and Daniel Marcu. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*, 139(1):91–107, 2002. ISSN 00043702. doi: 10.1016/S0004-3702(02)00222-9.
- Kevin Knight, Bianca Badarau, Laura Baranescu, Claire Bonial, Madalina Bardocz, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Martha Palmer, Tim O’Gorman, and Nathan Schneider. Abstract Meaning Representation (AMR) Annotation Release 2.0 LDC2017T10, 2017.
- Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. Neural AMR: Sequence-to-Sequence Models for Parsing and Generation. *Association for Computational Linguistics (ACL)*, pages 146–157, 2017. doi: 10.1145/nnnnnnnn.nnnnnnnn. URL <http://arxiv.org/abs/1704.08381>.
- Mahnaz Koupaee and William Yang Wang. WikiHow: A large scale text summarization dataset. *arXiv*, 2018. ISSN 23318422.

- Kundan Krishna and Balaji Vasani Srinivasan. Generating Topic-Oriented Summaries Using Neural Attention. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1697–1705, 2018. doi: 10.18653/v1/n18-1153. URL <http://aclweb.org/anthology/N18-1153>.
- Wojciech Kryściński, Romain Paulus, Caiming Xiong, and Richard Socher. Improving Abstraction in Text Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1808–1817, 2018. URL <http://arxiv.org/abs/1808.07913>.
- Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, 2020.
- Alex Kulesza and Ben Taskar. Learning Determinantal Point Processes in Sublinear Time. 2016. ISSN <null>. doi: 10.1109/GLOCOM.1997.638483. URL <http://arxiv.org/abs/1610.05925>.
- Yann LeCun and Yoshua Bengio. Convolutional Networks for Images, Speech, and Time-Series. *The handbook of brain theory and neural networks*, 1995.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. Technical report, 2020.

Chenliang Li, Weiran Xu, Si Li, and Sheng Gao. Guiding Generation for Abstractive Text Summarization Based on Key Information Guide Network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, volume 2, pages 55–60, 2018a.

Haoran Li, Junnan Zhu, Jiajun Zhang, and Chengqing Zong. Ensure the Correctness of the Summary: Incorporate Entailment Knowledge into Abstractive Sentence Summarization. *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1430–1441, 2018b. URL <https://aclanthology.info/papers/C18-1121/c18-1121>.

Haoran Li, Junnan Zhu, Jiajun Zhang, Chengqing Zong, and Xiaodong He. Keywords-guided abstractive sentence summarization. *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, (Clarke 2008):8196–8203, 2020. ISSN 2159-5399. doi: 10.1609/aaai.v34i05.6333.

Kexin Liao, Logan Lebanoff, and Fei Liu. Abstract Meaning Representation for Multi-Document Summarization. In *Proceedings of the 27th International Conference on*

- Computational Linguistics*, pages 1178–1190, 2018. URL <https://arxiv.org/pdf/1806.05655.pdf>.
- Rensis Likert. A technique for the measurement of attitudes. *Archives of psychology*, 1932.
- Chin-Yew Lin. Summary Evaluation Environment, 2001.
- Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL 2004*, number 1, pages 25–26, 2004. URL <papers2://publication/uuid/5DDA0BB8-E59F-44C1-88E6-2AD316DAEF85>.
- Chin-Yew Lin and Eduard Hovy. Manual and Automatic Evaluation of Summaries. In *Proceedings of the ACL-02 Workshop on Automatic Summarization-Volume 4*. Association for Computational Linguistics, 2002. URL <https://aclanthology.info/pdf/W/W02/W02-0406.pdf>.
- Hui Lin and Ja Bilmes. Learning mixtures of submodular shells with application to document summarization. *Uncertainty in Artificial Intelligence*, 2012. URL <http://arxiv.org/abs/1210.4871>.
- Hui Lin and Jeff Bilmes. A Class of Submodular Functions for Document Summarization. *Computational Linguistics*, 1:510–520, 2011. URL <http://ssli.ee.washington.edu/people/hlin/papers/lin-acl11-summ.pdf>.

- Junyang Lin, Shuming Ma, and Qi Su. Global Encoding for Abstractive Summarization. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 163–169, 2018. doi: 10.1111/jcpp.12768. URL <https://www.github>.
- Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A Smith. Toward Abstractive Summarization Using Semantic Representations. In *Proceedings of the 2015 Conference of the NAACL HLT*, pages 1077–1086, 2015.
- Yang Liu and Mirella Lapata. Text Summarization with Pretrained Encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, aug 2019. URL <http://arxiv.org/abs/1908.08345>.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv*, jul 2019. URL <http://arxiv.org/abs/1907.11692>.
- Annie Louis and Ani Nenkova. Automatically Assessing Machine Summary Content Without a Gold Standard. *Computational Linguistics*, 39(2):267–300, 2013. ISSN 01272713. doi: 10.1162/COLI. URL <http://www.seas.upenn.edu/>.
- Jordan J Louviere, Terry N Flynn, Anthony Alfred Fred, and John Marley. *Best-worst scaling: Theory, methods and applications*. Cambridge University Press, 2015.

- Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421. The Association for Computational Linguistics, 2015.
- Christian Matthiessen and John A Bateman. *Text generation and systemic-functional linguistics: experiences from English and Japanese*. Pinter Publishers, 1991.
- Jonathan May. SemEval-2016 Task 8: Meaning Representation Parsing. *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1063–1073, 2016.
- Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.173. URL <https://aclanthology.org/2020.acl-main.173>.
- Ryan McDonald. A Study of Global Inference Algorithms in Multi-Document Summarization. *Advances in Information Retrieval*, 5478:806–809, 2009. ISSN 19454589. doi: 10.1007/978-3-642-00958-7. URL <http://www.springerlink.com/content/0w01373601n982j7>.
- Kathleen R. McKeown, Judith L Klavans, Vasileios Hatzivassiloglou, Regina Barzilay, and Eskin Eleazar. Towards multidocument summarization by reformulation: Progress and Prospects. In *Proceedings of the 16th National Conference on American Association*

for Artificial Intelligence, pages 453–460, 1999. ISBN 0-262-51106-1. URL <http://www.aaai.org/Papers/AAAI/1999/AAAI99-065.pdf>.

Afonso Mendes, Shashi Narayan, Sebastião Miranda, Zita Marinho, André F. T. Martins, and Shay B. Cohen. Jointly extracting and compressing documents with summary state representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, US, 2019.

Rada Mihalcea and Paul Tarau. TextRank: Bringing order into texts. *Proceedings of EMNLP*, 85:404–411, 2004. ISSN 0256307X. doi: 10.3115/1219044.1219064. URL <http://acl.ldc.upenn.edu/acl2004/emnlp/pdf/Mihalcea.pdf>.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. Technical report, 2013. URL <https://arxiv.org/pdf/1310.4546.pdf>.

Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos Santos, Caglar Gulcehre, and Bing Xiang. Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond. *Proceedings of CoNLL*, pages 280–290, 2016. URL <http://arxiv.org/abs/1602.06023>.

Shashi Narayan, Ronald Cardenas, Nikos Papasarantopoulos, Shay B Cohen, Mirella Lapata, Jiangsheng Yu, and Yi Chang. Document Modeling with External Attention for Sentence Extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 2020–2030, Melbourne, Australia, 2018a.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. Ranking Sentences for Extractive Summarization with Reinforcement Learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 1747–1759, Stroudsburg, PA, USA, 2018b. Association for Computational Linguistics. URL <http://aclweb.org/anthology/N18-1158>.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't Give Me the Details, Just the Summary! Topic-aware Convolutional Neural Networks for Extreme Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018c.

Shashi Narayan, Yao Zhao, Joshua Maynez, Gonçalo Simões, Vitaly Nikolaev, and Ryan McDonald. Planning with Learned Entity Prompts for Abstractive Summarization. *Transactions of the Association for Computational Linguistics*, 9:1475–1492, 12 2021. ISSN 2307-387X. doi: 10.1162/tacl.a.00438. URL <https://doi.org/10.1162/tacl.a.00438>.

Ani Nenkova and Rebecca Passonneau. Evaluating content selection in summarization: The pyramid method. *Proceedings of HLT-NAACL*, 2004:145–152, 2004. URL <http://www.isi.edu/papers2://publication/uuid/DC675E84-0A45-48B7-A26C-F08B4B9398D3>.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajič, Angelina Ivanova, and Yi Zhang. SemEval 2014 task 8: Broad-coverage semantic dependency parsing. In *Proceedings of the 8th International Workshop on Semantic*

- Evaluation (SemEval 2014)*, pages 63–72, Dublin, Ireland, August 2014. Association for Computational Linguistics. doi: 10.3115/v1/S14-2008. URL <https://aclanthology.org/S14-2008>.
- You Ouyang, Wenjie Li, Sujian Li, and Qin Lu. Applying regression models to query-focused multi-document summarization. *Information Processing and Management*, 47(2):227–237, 2011. ISSN 03064573. doi: 10.1016/j.ipm.2010.03.005. URL <http://dx.doi.org/10.1016/j.ipm.2010.03.005>.
- Ramakanth Pasunuru and Mohit Bansal. Multi-Reward Reinforced Summarization with Saliency and Entailment. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 646–653, 2018. doi: 10.18653/v1/N18-2102. URL <http://aclweb.org/anthology/N18-2102><http://arxiv.org/abs/1804.06451>.
- Romain Paulus, Caiming Xiong, and Richard Socher. A Deep Reinforced Model for Abstractive Summarization. In *Proceedings of the 6th International Conference on Learning Representations*, 2018. URL <http://arxiv.org/abs/1705.04304>.
- Laura Perez-Beltrachini and Mirella Lapata. Multi-Document Summarization with Determinantal Point Process Attention. *Journal of Artificial Intelligence Research*, 71:371–399, 2021. ISSN 10769757. doi: 10.1613/jair.1.12522.
- Matthew E Peters, Mark Neumann, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational*

- Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, 2018. URL <http://allennlp.org/elmo>.
- Maxime Peyrard and Iryna Gurevych. Objective Function Learning to Match Human Judgements for Optimization-Based Summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 654–660, 2018. doi: 10.18653/v1/N18-2103. URL <http://tac.nist.gov/2008/>.
- Matt Post and David Vilar. Fast Lexically Constrained Decoding with Dynamic Beam Allocation for Neural Machine Translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018. URL <https://awsllabs.github.io/sockeye/>.
- Nima Pourdamghani, Kevin Knight, and Ulf Hermjakob. Generating English from Abstract Meaning Representations. In *Proceedings of the 9th International Natural Language Generation*, volume 0, pages 21–25, 2016. URL <http://www.isi.edu/natural-language/mt/inlg-16.pdf>.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving Language Understanding by Generative Pre-Training. Technical report, OpenAI, 2018. URL <https://gluebenchmark.com/leaderboard>.

- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.
- Pengjie Ren, Furu Wei, and Zhumin Chen. A Redundancy-Aware Sentence Regression Framework for Extractive Summarization. pages 33–43, 2016.
- D. W. Robertson. A Note on the Classical Origin of ” Circumstances ” in the Medieval Confessional. *Studies in Philology*, 43(1):6–14, 1946.
- Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object hallucination in image captioning. *arXiv preprint arXiv:1809.02156*, 2018.
- Alexander M Rush, Sumit Chopra, and Jason Weston. A Neural Attention Model for Abstractive Sentence Summarization. *In Proceedings of the Conference on EMNLP*, (September):379–389, 2015. ISSN 19909772. doi: 10.1162/153244303322533223. URL <http://arxiv.org/abs/1509.00685>.
- Shinsaku Sakaue, Tsutomu Hirao, Masaaki Nishino, and Masaaki Nagata. Provable Fast Greedy Compressive Summarization with Any Monotone Submodular Function. *In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1737–1746, 2018. doi: 10.18653/v1/n18-1157. URL <http://aclweb.org/anthology/N18-1157>.
- Evan Sandhaus. The New York Times Annotated Corpus, 2008.

- Natalie Schluter. The limits of automatic summarisation according to ROUGE. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Short Papers*, pages 41–45, Valencia, Spain, 2017.
- Abigail See, Peter J Liu, and Christopher D Manning. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the ACL*, 2017. ISBN 9781945626753. doi: 10.18653/v1/P17-1099. URL <http://arxiv.org/abs/1704.04368>.
- Elaheh ShafieiBavani, Mohammad Ebrahimi, Raymond Wong, and Fang Chen. Summarization Evaluation in the Absence of Human Model Summaries Using the Compositionality of Word Embeddings. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 905–914, 2018. URL <https://aclanthology.info/papers/C18-1077/c18-1077>.
- R R Sokal and F J Rohlf. *Biometry*. 3rd ed. 1995.
- Kaiqiang Song, Lin Zhao, and Fei Liu. Structure-Infused Copy Mechanisms for Abstractive Summarization. *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1717–1729, 2018. URL <https://aclanthology.info/papers/C18-1146/c18-1146>.
- Linfeng Song, Yue Zhang, Xiaochang Peng, Zhiguo Wang, and Daniel Gildea. AMR-to-text generation as a Traveling Salesman Problem. In *Proceedings of the 2016 Conference on EMNLP*, pages 2084–2089, 2016. URL <http://arxiv.org/abs/1609.07451>.

- Si Sun, Chenyan Xiong, Zhenghao Liu, Zhiyuan Liu, and Jie Bao. Joint keyphrase chunking and salience ranking with BERT. *arXiv*, 2020. ISSN 23318422. URL <http://arxiv.org/abs/2004.13639>.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems (NIPS)*, pages 3104–3112, 2014. ISSN 09205691. doi: 10.1007/s10107-014-0839-0. URL <http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural>.
- Simone Teufel and Hans Van Halteren. Evaluating information content by factoid analysis: Human annotation and stability. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 419–426, 2004.
- L. L. Thurstone. A Law of Comparative Judgment. *Psychological review*, 101(2):255–270, 1994.
- Rik Van Noord and Johan Bos. Neural semantic parsing by character-based translation: Experiments with abstract meaning representations. *Computational Linguistics in the Netherlands Journal*, 7(2016):93–108, 2017. ISSN 22114009.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 2017-Decem, pages 5999–6009. Neural information processing systems foundation, jun 2017.
- Andreas Vlachos and Sebastian Riedel. Fact Checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 Workshop on Language Technologies and*

Computational Social Science, pages 18–22, Baltimore, MD, USA, jun 2014. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/W14-2508>.

Xuanhui Wang, Cheng Li, Nadav Golbandi, Michael Bendersky, and Marc Najork. The LambdaLoss framework for ranking metric optimization. *International Conference on Information and Knowledge Management, Proceedings*, pages 1313–1322, 2018. doi: 10.1145/3269206.3271784.

Sam Wiseman, Stuart M Shieber, and Alexander M Rush. Challenges in Data-to-Document Generation. In *Proceedings of the 2017 Conference on EMNLP*, pages 2253–2263, Copenhagen, Denmark, 2017. doi: 10.18653/v1/D17-1239. URL <http://www.aclweb.org/anthology/D17-1239><https://arxiv.org/pdf/1707.08052.pdf>.

Jordan J Louviere Woodworth and George G. Best-worst scaling: A model for the largest difference judgments. 1991.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. Technical report, 2016. URL <http://arxiv.org/abs/1609.08144>.

- Shudong Yang, Xueying Yu, and Ying Zhou. LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example. *Proceedings - 2020 International Workshop on Electronic Communication and Artificial Intelligence, IWECAI 2020*, pages 98–101, 2020. doi: 10.1109/IWECAI50956.2020.00027.
- Yinfei Yang, Forrest Sheng Bao, and Ani Nenkova. Detecting (Un)Important Content for Single-Document News Summarization. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 707–712, 2017. URL <https://aclanthology.info/papers/E17-2112/e17-2112><http://arxiv.org/abs/1702.07998>.
- Yasuhisa Yoshida, Jun Suzuki, Tsutomu Hirao, and Masaaki Nagata. Dependency-based Discourse Parser for Single-Document Summarization. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1834–1839, 2014. URL <http://www.aclweb.org/anthology/D14-1196>.
- Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*. PMLR, 2020.
- Jingyi Zhang, Masao Utiyama, Eiichiro Sumita, Graham Neubig, and Satoshi Nakamura. Guiding Neural Machine Translation with Retrieved Translation Pieces. In *Proceedings of the 16th Annual Conference of the NAACL HLT*, New Orleans, Louisiana, apr 2018. The Association for Computational Linguistics. URL <http://arxiv.org/abs/1804.02559>.

Zheng Zhao, Shay B. Cohen, and Bonnie Webber. Reducing quantity hallucinations in abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2237–2249, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.203. URL <https://aclanthology.org/2020.findings-emnlp.203>.