Extractive Summarization of Scientific Publications

Krušić, Lucija

University of Rijeka - Department of Informatics

Diplomski studij engleskog jezika i književnosti i Informatike

Lucija Krušić

Extractive summarization of scientific publications

Master's thesis

Mentor: prof. dr. sc. Sanda Martinčić-Ipšić, dipl. ing.

Rijeka, 9th of September 2019

Abstract

This Master's thesis aims to provide a comprehensive survey of state-of-the art methods used for the task of Automatic Text Summarization. Automatically made summaries can provide great benefits to everyday internet users and enhance the way we search for relevant and necessary information and save time and resources invested in human-made summaries. This thesis provides an overview of the field of ATS and covers the approaches to summarization, the real-world applications of summarization and the various evaluation metrics used to establish the quality of the generated summary. Automatic text summarization is a blooming field which has recently gained significant interest and presently, much advancement is being achieved with the use of neural networks. The thesis will provide a comprehensive survey of the recent work done in the field, including the datasets used and the state-of-the art results of recent studies. The various methods of approaching ATS are described in depth and compared on the basis of their effectiveness.

Keywords:

automatic text summarization, extractive summarization, abstractive summarization, natural language processing, deep neural networks

Sveučilište u Rijeci - Odjel za Informatiku

Diplomski studij engleskog jezika i književnosti i Informatike

Lucija Krušić

Ekstraktivna sumarizacija znanstvenih radova

Diplomski rad

Mentor: prof. dr. sc. Sanda Martinčić-Ipšić, dipl. ing.

Rijeka, 9. rujna 2019.

Sažetak

Ovaj diplomski rad pruža opsežan pregled najsuvremenijih metoda koje se koriste u polju Automatskog Sažimanja Teksta. Automatski generirani sažeci donose velike prednosti svakodnevnim korisnicima internet te u znatnoj mjeri reduciraju vrijeme i resurse uložene u stvaranje ljudski kreiranih sažetaka. Ovaj rad će pružiti pregled samoga polja i obuhvatiti različite vrste automatskog sažimanja te pružiti uvid u brojne primjene automatskog sažimanja u stvarnome svijetu. Osim toga, biti će opisane metode za evaluaciju generiranih sažetaka koje su se pokazale najprihvatljivijima za određivanje kvalitete generiranih sažetaka. Automatsko sažimanje teksta je polje Računalne obrade prirodnog jezika koje trenutno proživljava novi procvat a napretci u polju se postižu gotovo svakodnevno a osobito u zadnje vrijeme sa sve češćim korištenjem neuronskih mreža za različite zadatke koji spadaju pod obradu prirodnog jezika. Ovaj diplomski rad pružiti će pregled najnovijih istraživanja, što uključuje i zbirke koje se koriste za zadatak automatske sumarizacije kao i najsuvremenije i najuspješnije modele te njihove rezultate. Raznolike metode pristupanja automatskom sažimanju teksta su detaljno opisane i uspoređene na temelju njihove funkcionalnosti i uspješnosti.

Ključne riječi:

Automatsko sažimanje teksta, ekstraktivno sažimanje, apstraktno sažimanje, računalna obrada prirodnog jezika, duboke neuronske mreže

Contents

1. Introduction

In the last two decades, we have witnessed a dramatic increase in the amount of information - both textual and conversational, that correlated with the rapid growth in data and the rise of Internet. Unlike in the past, when the main sources of information were books, the majority of information nowadays is stored digitally. Furthermore, prior to the development of the Web and the "Internet revolution" the vast majority of conversation was in spoken form. In recent times, that has been changing dramatically – with the increased usage of web-based forms of communication, such as e-mail, instant messaging, conversational agents, blogs and social networks. At the same time, we have seen the astounding development of speech technologies and ASR systems that allow us to automatically transcribe meetings, phone conversations or other forms of spoken communication (Carenini, Murray, & Ng, 2011).

Consequently, all kinds of information, including human conversational data, are accumulating in an astonishing speed. Due to the large amounts of data, manually creating summaries from long documents is proving to be a time-consuming, budget-inefficient and an extremely complex task (the experts need to be qualified and unbiased); therefore, there is a demand for the development of tools that can allow us to create summaries of any type of text automatically (Dong, 2018). To be able to manage and to use information in an optimal way, there has been an increased development of summarization techniques used to extract information from large amounts of data. It is both exhausting and time-consuming to read large amounts of text and it leads to both neglecting the vital details and consuming redundant information. Therefore, nowadays, there is a necessity to ease the users' acquisition of information.

Condensing the most important information in form of a summary would benefit a large number of users, as it could allow us to quickly find the appropriate literature, when writing a review or a scientific text, to skim through a long email thread with ease and be able to join an ongoing conversation. Furthermore, the applications of automatic summarization prove to be beneficial in providing information retrieval and recommendation systems, since the ability to successfully and precisely summarize documents can produce search results of a higher quality.

Further applications can be found in the fields of business intelligence to preserve corporate memory, forensic investigation and to analyze large-scale trends and to monitor public opinion – which is becoming increasingly important with the exchange of opinion and news on social media, especially Twitter (Carenini, Murray, & Ng, 2011).

Nowadays, research on text summarization attempts at improving the level of abstractiveness of generated summaries, as to make the summaries appear as human-made as possible. Abstractive summarization refers to the process of deriving knowledge from the original text and leveraging it to create novel sentences that do not appear in the original, but still convey the most important information of the original text. Unlike abstractive summarization, which is a complex and daunting task, a much simpler form of ATS is extractive summarization. Extractive summarization is the process of automatically creating a summary by combining the most salient sentences extracted from the original text into a more concise text (Dong, 2018). Some of the first work on automatic summarization dealt with the task of extractive summarization.

Automatic text summarization (ATS) or the art of summarizing the given content into smaller texts has been a field of interest much prior to the birth of Internet. The idea of automatic text summarization became widely researched in the 50s, following the publication of Luhn's "The automatic creation of literature abstracts" (1958). Since then, there has been much development of techniques and approaches in this field of research. However, automatic summarization is nevertheless a strenuous task, as there are many complex issues arising when attempting to summarize a larger number of documents and produce a high-quality summary. Some of those issues pertain to topic labeling and topic segmentation, which refer to identifying which portions of the text deal with the same topic and generating informative labels for each topic found in the original text. Furthermore, a challenging task is opinion mining when it comes to more subjective texts, such as conversations (e-mail, text messages and transcriptions of conferences). Perhaps one of the most challenging aspects of attempting to automatically summarize documents is the process of selecting the most relevant and essential content from one document and how to generate condensed content in a way that it is informative or indicative and useful to the end user (Saggion & Poibeau, 2013). The high-quality summary needs to consist of the most important information in the text for it to be useful and non-redundant, cohesive, short and significant.

This Master's thesis aims to overview the methods used for automatic text summarization and apply the selected method to preform extractive summarization of scientific texts in the English language. The corpus of scientific texts used for this task will consist of research papers from the domain of summarization. The summaries created through the process of extractive summarization will be compared with the summaries provided by the authors of the scientific papers used in the process. The thesis will provide a review of research in the field of summarization and delve into state of the art methods used to achieve abstractive and extractive summarization of texts. Furthermore, the thesis will provide an overview of evaluation metrics used for assessing the quality of summaries. The goal of this thesis is to provide both a useful overview of the technologies used for automatic summarization and advances in the field as well as to provide insights into what are the possible future developments of the field.

In the second chapter the general overview of the field of ATS will be provided, as well as the types of ATS and the current and the possible real world applications of the field. Furthermore, the third chapter will revolve around the approaches to automatic text summarization and an overview of how the field developed in the recent decades, more precisely the methods of achieving automatic text summarization that have been popular in the past and nowadays with a focus on the usage of neural networks for creating automatically generated summaries. The most relevant and widely used evaluation metrics will be discussed in chapter four. The fifth chapter will provide a comprehensive survey of recent studies in the field of ATS and the state-of-the-art methods. The statistical evaluation of the recent research and the possible gaps in the field will be discussed in chapter five.

2. Automatic text summarization

The field of automatic text summarization has stemmed from the need to develop techniques which would create conventional abstracts by automatic means (Luhn, 1958). The process of writing abstracts proved to be both time-consuming and an intellectual effort, which demands a familiarity with the topic. Furthermore, the abstract tends to be influenced by the abstracter's attitude towards the topic, their background, capabilities, opinion and interests. That leads to the lack of reliability; since the same human abstracter might not produce the same abstract if written on different occasions and two abstracters might differently interpret the writer's word, therefore leading to different abstracts hence summarization is a highly subjective task. Consequently, the reader will greatly benefit from an unbiased, automatically-made abstract.

With time, the definition of automatic text summarization remains unchanged, the process of automatically producing a concise and fluent summary while preserving the key information, concepts and overall meaning (Allahyari, 2017). The basic tasks on which all summarization systems work have remained the same: to construct an intermediate representation of the input text which expresses the main aspects of the text, to score the sentences based on representation and to select a summary comprising of a number of sentences (Allahyari, 2017).

However, the techniques used for the task have developed greatly with the last decades, as have the evaluation metrics used to assess the quality of the summary by comparing it to human-made summaries. With the development of technology, automatic text summarization has found various new applications and one again became a booming field of research.

Various areas of the field of summarization and summarization tasks can be identified through investigating the factors such as summarization input, purpose and output or the way in which a summary is generated, how it is presented to the user and the function of the intended summary (Carenini, Murray, & Ng, 2011).

2.1. Types of summarization

When it comes to the input of summarization, we can categorize summaries into single-document and multi-document summaries (Carenini, Murray, & Ng, 2011) (Gupta & M., 2016). The output of the single- document summary can be an abstract or an outline of a certain article or just its headline, or the input to the task of summarization can be a single email. When it comes to the multi-document summary, the input can consist of multiple articles with a similar topic, an email thread, etc. Consequently, multi-document summary is a far more complex task because it raises the issue of redundancy, which has been tackled by various approaches (such as MMR[1] approach) (Carenini, Murray, & Ng, 2011) (Gupta & M., 2016). When it comes to summarizing text conversations, hybrid approaches have been devised, which base on summarizing preceding conversation to provide most suitable context (Carenini, Murray, & Ng, 2011).

Moreover, the input of summarization on the basis of language can be monolingual, multilingual and cross-lingual. A monolingual summarization system is one in which the input and the output document are in the same language. When the input files are in multiple languages (e.g. Croatian, German, English and Spanish) and the files produced through the process of summarization are summaries in all of those languages, we are referring to a multi-lingual summarization task. A cross-lingual summarization task refers to the process in which the language of the source document is different than the language of the target document or summary, e.g. if the source document was written in Swedish and the summary produced was in any other language other than Swedish, such as English (Gupta & M., 2016).

Further categorization of summarization is based on the purpose of the summarization process, meaning if the objective of the summary is giving general content of a document or responding to a user query. A query-based summary (also named query-focused, user-focused or topic-focused) includes information related or responding to a user provided query, which means that there needs to be a significant overlap between the content of the summary and the query. When

[1] MMR (Maximal Marginal Relevance) – a method for re-ranking sentences during sentence selection and incrementally creating an automatic summary

it comes to generic summaries, they have no specific purpose other than providing the user the general, most important information gathered from the original text.

Summaries can also be distinguished on the basis their intended function with regards to the source document (Carenini, Murray, & Ng, 2011). The purpose of the summary can be either informative or indicative. Informative summaries are generated with the purpose to convey the most important information in the source document while indicative summaries represent the main idea, or the high-level outline of the source document and are supposed to serve as a starting point for the reader (to decide whether or not they are interested in reading the document). Indicative summaries are an outline of the original text, conveying only the critical information from the source. Their purpose is for the user to decide whether or not they want to read the original document. Usually, automatically generated summaries can be a mixture of both indicative and informative summaries (Carenini, Murray, & Ng, 2011).

When it comes to the type of output generated by the summarization process, summarization can be classified into abstractive and extractive (Dong, 2018). Extractive summarization refers to the process of generating an exact summary by extracting the most relevant sentences from the source document. That can be achieved by computing the measure of informativeness for each sentence and then selecting and ordering the most significant sentences. Informativeness can be defined as the ability of a sample to reduce the generalization errors of the classification model (Du, Wang, Zhang, Zhang, & Lieu, 2015). The informativeness of each item is measured against the query parameters. This type of summarization is a simpler and a more robust than abstractive summarization. Abstractive summarization tends to be a more complex and challenging task, considering that it is a process of creating novel sentences that would come as close as possible to a human-written summary. An abstractive summarizer works by extracting information from the source document, deriving novel knowledge by inference or aggregating and abstracting knowledge and lastly, selecting the most informative content to be a part of the summary (Carenini, Murray, & Ng, 2011). Between the two types of summarization, extractive summarization has been far more popular in research due to the fact that it is by far easier to implement. However, abstractive summaries remain the ultimate goal of summarization, as we strive to produce the most human-like summary.

Further categorization of summarization can be found in modes of communication as well as domains to which the source text belongs. Solutions and solutions can be created for a specific domain or for general purpose. When solutions are generally applicable for any type of text in any domain, they tend to be of lesser quality and precision than the highly effective, domain-specific tools. However, a multi-modal approach that would take summarize texts from different modalities and texts spanning through different modalities by taking advantage of the features shared by all the modalities and facilitating domain adaptation could be an optimal solution (Carenini, Murray, & Ng, 2011).

The final category with regards to type of summarization output would be textual and multimedia output. An automatically generated summary can be presented in a form of a textual document, but also as an illustration or a graphic diagram. The output could combine both media and be in a form of a word cloud, in which visual representation and a textual summary complement each other (Carenini, Murray, & Ng, 2011).

2.2. Applications of ATS

Even though some of the applications of summarization systems have been briefly discussed in the Introduction of this paper, we will delve a bit deeper into the topic. One of the first applications that the development of summarization methods was intended for was the summarization of scientific texts (Luhn, 1958). The purpose of this was to ease the reader's research process by providing the reader with automatically created abstracts which would be valid and reliable and save on the manpower which would be used to manually write abstracts. This kind of abstract would be unbiased and free of the influence of the abstracter's background, attitudes towards the topic and therefore of an invariable quality.

Nowadays, with the development of social networks, blogs, the growth of email and the speech technology revolution, the summarization applications seem limitless. One of the interesting ways that summarization could make our day-to-day conversations more convenient would be joining an ongoing conversation on a forum, a micro-blog (such as Twitter) or a blog. Instead of reading through the whole thread, summarization would allow us to read the most important comments and the prevalent opinions in a short format and to be able to make an informed comment. This could also be applied to the situation of joining an email thread (or discussions in a forum) with various participants – summarization tools can be an enormous aid to making more informed decisions, learning more vital and detailed information and staying on top of your correspondences.

This would be a great asset for business as well, especially for the situation of joining an ongoing meeting or an ongoing email thread. Further applications to business could be preserving corporate memory and optimizing decision-making practices, evaluating employee effectiveness and communication patterns. It could help the workflow of internal documents, which would help assess and reuse previous information belonging to a certain topic and quickly assemble reports. When it comes to banking firms and stock trading, summarization tools can significantly help the analysis of market reports and financial news which would be a great asset for such companies.

Automatic summarization systems can also be of great help for marketing firms as they are able to summarize large conversations online, which would aid with analyzing large scale trends. Twitter and micro-blogs have nowadays become one of the most relevant platforms for news-

sharing and the exchange of opinion and summaries of such conversations can provide an overview of people's opinions (Carenini, Murray, & Ng, 2011)

Moreover, summarization systems can also prove to be an invaluable asset to medicine and helping the disabled as it would allow quicker analysis of medical cases as well as the improvement of voice-to-text technologies designed for people with hearing disabilities (Ratia, 2018). The ability to quickly and automatically summarize text conversations can be a great help to forensic investigation as nowadays, emails and text messages are also viable evidence in the court of law. Furthermore, summarization tools can be of great help for question-answer bots and could, potentially, facilitate the replacement of humans by artificial intelligence when it comes to automated content creation (Ratia, 2018).

The applications of ATS are great and with the development of new technologies, they continue expanding. For that reason, the development of new and improved techniques for automatic summarization of speech and text is a priority and a booming field of research.

3. Approaches to summarization

Nowadays, the most effective and popular summarization techniques are neural-based techniques. However, in this thesis, methods used prior to the "neural-network era" (Gupta & M., 2016) will also be briefly described.

3.1. Approaches prior to the neural-network era

The first attempts to text summarization were based on statistical approaches, which deal with statistical features designed to establish the importance of sentences and words in a text. Statistical approaches are independent of a particular language and do not require any additional linguistic knowledge or linguistic processing. Therefore, they can be used on any text in any language for the purpose of creating a summary.

Some of the early work was based on previously mentioned article by Luhn (The Automatic Creation of Literature Abstracts, 1958), who proposed an algorithm that defined the sum of frequencies of significant words by ignoring all the stop words (high-frequency words from closed classes - stopwords), determining and selecting top words (the most occurring open classes words in the text) and selecting top sentences which are scored according to the amount of top words they contain. Luhn worked on creating abstracts from scientific papers and magazine articles by using statistical methods which were the basis of all further research.

Further developments were based on the Edmundsonian paradigm (Edmunson, 1969), which refers to ranking each sentence in relation to other sentences so that the highest ranked sentences are extracted and form a summary (Afantenos, Karkaletsis, & Stramatopoulous, 2005).

Moving forward, some of the techniques included the reimplementation of Luhn's algorithm but through the latent Dirichlet model of allocation (LDA) which is a generative probabilistic model for collections of discreet data such as text corpora (Blei, 2003). LDA is based on a Bayesian model in which each item of a collection is modeled as a finite mixture over an underlying set of topic probabilities (Blei, 2003). In this approach, instead of the frequency of words, topic probabilities are used to represent a textual document. The LDA is trained on a certain type of

text, the topic distribution is determined and inferred for each sentence. Lastly, the most relevant sentences from the topics which are dominant in the text are extracted and form a summary.

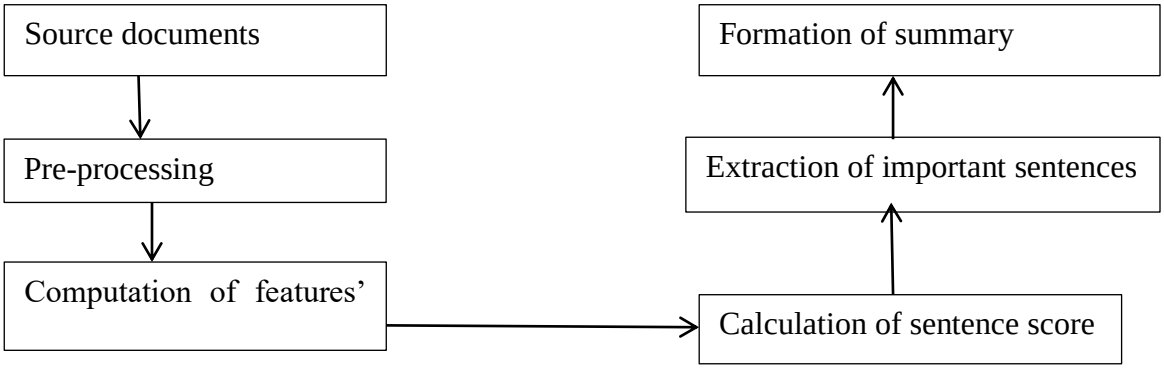


Figure 1 – Block diagram of summarization by using statistical techniques (Gupta & M., 2016)

Usually in these approaches, weights need to be awarded to each word. Some of the techniques used for achieving that are frequency driven approaches – they can be based on the probability of words and the TFIDF (Term Frequency Inverse Document Frequency) representation model. Word probability is simply the number of occurrences of a word *f(w)*, divided by the number of all the words in a text.

$$P(w) = \frac{f(w)}{N} \tag{1}$$

Term Frequency Inverse Document Frequency is a method which ignores the stop words in a text and assesses the importance of words and reduces the influence of frequent words by lowering their weights.

$$q(w)=f_d(w)*\log\frac{|D|}{f_{D(w)}} \tag{2}$$

Another statistical approach to creating summaries automatically is using the BOW (bag-of-words) approach. Using the simplified BOW method, sentences are represented as unordered collections of n-grams, more specifically unigrams (Carenini, Murray, & Ng, 2011).

The BOW systems do not use lists or hand-crafted dictionaries but rather lexical features such as n-grams while using a machine learning method to establish the extent of correlation of the features with positive and negative classes (Carenini, Murray, & Ng, 2011).

The n-gram language model is one of the simplest models for assigning probabilities to sequences of words (Jurafsky & Martin, 2018). N-grams can be defined as sequences of co-occurring words within a given window. If we consider a sequence consisting of two words such as “the cat”, such n-gram would be donned a bigram, while sequences of three words are called trigrams. A sequence of a single word is named a unigram (Banko & Vanderwende, 2004).

Methods such as TextRank, (Mihalcea, 2004) influenced by the PageRank algorithm, represent the documents as a graph. The similarity of the sentences is commonly measured by connecting two vertices, as the sentences form vertices of the graph and the edges between sentences serve to demonstrate how similar the two sentences are. Sentences that are connected to many other sentences could have higher centrality value (degree, closeness, beetweenes, page rank, eigenvector, etc.) which makes them more relevant and therefore more likely to be part of the summary. TextRank is a graph-based model in which the text is first pre-processed by performing part-of-speech tagging and lemmatization for every sentence of the document following extracting the key phrases along with their weights. A score is calculated for each sentence based on the jacquard distance between the sentences and the key phrases, which then form a summary.

Another graph based method that was proposed was GraphSum (Baralis, Cagliero, Mahoto, & Fiori, 2013) which firstly ranks the combinations of two or more terms using the PageRank strategy and with the node ranking produced, selects the sentences used for the summary. This algorithm prevents the neglect of certain words, as previous graph based methods did.

Other, more advanced methods, included machine learning models such as the Naïve Bayes and the Support Vector Machines (SVM) were used for the classification of sentences, after the extraction of surface, content and relevance features for the sentence representation (Dong, 2018). The Probabilistic Support Vector Machine was employed as supervised learning and the Naïve Bayes Classifier in combination with PSVM was used for semi-supervised learning in Wong, Wu, Li (2008) for the purpose of categorizing sentences by relevance, surface, content

and event features. They claimed that the use of semi-supervised learning saves time and cost for labelling and that combining the features improves the quality of the summarization.

Furthermore, Hidden Markov Models were used to determine the likelihood of appearance of each sentence in a summary. This technique proved to be more successful than the ones previously used, in comparison to the human-made summary (Conroy & O'leary, 2001).

All these methods were mainly used for the task of extractive summarization. While the task of abstractive summarization is far more demanding, because it entails the generation of novel language, attempts were made at achieving that goal. Reiter and Dale (1997) used Natural Language Generation techniques to develop a system that would generate text. It was done by identifying the main ideas in documents and encoding them into feature representations (Dong, 2018)

Other early work on abstractive summarization was based on a semi-manual process of locating the main ideas of the texts and using prior knowledge (scripts and templates) to produce summaries by slot filling and smoothing techniques (Dong, 2018).

3.2 Neural-network based approaches

With the recent developments in deep learning methods, the main techniques used for automatic summarization have been based on neural networks. Deep learning methods and algorithms have produced substantial advances in fields such as computer vision, which enticed researchers to attempt to use deep learning methods, among which neural networks have proven to be the most successful, for various natural language processing tasks (Young, Harazika, Poria, & Cambria, 2018).

Neural-based methods depend on the amount of the training data, which if sufficient, can provide better summaries with less human involvement. Research by Collobert (2011) investigated the performance of a simple neural network architecture combined with a learning algorithm on various NLP tasks, such as part-of-speech tagging, chunking, name-entity recognition and semantic role labeling. The purpose of the research was to compare the acquired results with the results gained from previously state-of-the art methods (which applied linear statistical models to ad-hoc features). Previously, the features were engineered to be task-specific, derived from the output of pre-existing systems and leveraged linguistic knowledge. However, Collobert (2011) as well as further research has been inclined to the usage of neural network architectures, as they provide superior results and allow us to get closer to achieving the goal of natural language understanding as well as the broader goals of artificial intelligence.

Neural networks are based on transforming the words into continuous vectors (word embeddings), encoding the sentences or documents as continuous vectors and representing sentence or documents which are then given to the model to be either selected (when it comes to extractive summarization) or generated (when it comes to abstractive summarization) (Dong, 2018). During the stage of transformation of words into vectors, neural networks can be used to obtain pre-learned lookup tables (such as in Word2Vec, CW Vectors and GloVe (Dong, 2018)). Furthermore, both convolutional neural networks (CNNs) and recurrent neural networks (RNNs) can be used as encoders for extracting features and neural network models can be used as regressors for selection (extraction) or decoders for generation (abstraction) (Dong, 2018).

3.2.1. Word embedding models

Word embeddings, also called distributional vectors, are learned representations of text where words that have a similar meaning tend to occur in a similar context, therefore have similar vector representations (Young, Harazika, Poria, & Cambria, 2018). Words (phrases, sentences, or documents) are mapped into low-dimensional vectors of real numbers - embeddings.

Following the distributional hypothesis, the purpose of distributional vectors is to reveal the characteristics of neighboring words and discover the similarity between words. According to (Young, Harazika, Poria, & Cambria, 2018) word embeddings are usually pre-trained by optimizing an auxiliary objective in a large unlabeled corpus which results the ability of learned words to capture information on syntax and semantics and to predict a world based on its context. That is what makes them highly successful in a large variety of NLP tasks.

According to (Goldberg, Neural Network Models for Natural Language Processing: Synthesis Lectures on Human Language Technologies, 2017) a benefit of using low-dimensional vectors is that the majority of toolkits do not operate well with high-dimensional, sparse vectors. The greatest advantage of dense vectors is generalization, since it provides us with a representation that allows us to discover all the similarities between words (Goldberg, Neural Network Models for Natural Language Processing: Synthesis Lectures on Human Language Technologies, 2017). Therefore, the process of embedding is used to transform a space with many dimensions per word to a continuous vector space with lower dimensions which can be done by the use of neural networks, applying dimensionality reduction techniques on the word occurrence matrix and by the use of probabilistic models.

The techniques used for dimensionality reduction can be developed in the area of distributional semantics and in the area of language modeling, although these methods are highly interconnected. Both create word embeddings from low-rank factors of a co-occurrence matrix whose elements are the frequency of seeing words together; however when it comes to distributional semantics the factorization of the co-occurrence matrix is done explicitly, while in the area of language modeling it is performed implicitly (Ljungberg, 2017). One of those explicit methods is PSA (Principal Component Analysis) which is a linear transformation technique and

a standard method for feature space reduction. PCA is an unsupervised method as it disregards class labels and searches for the principal components that maximize variance in a dataset. The other hand, LDA (Linear Discriminant Analysis), another linear transformation technique, is considered to be a supervised algorithm since it computes the linear discriminants that represent the axes which serves to maximize the separation between multiple classes. The purpose of this process is to obtain the optimal feature subspace, or that the values of the vectors in the matrix have similar magnitudes (Raschka, 2014).

Pre-trained embeddings are obtained from a large corpus of texts through algorithms which are based on the distributional hypothesis. An example of "good vectors" or pre-trained embeddings are the dense vector representations of the learned vector for the word "dog" and its similarity to the learned vector of the word "cat". If there are enough occurrences of the word "cat" so that its vector is similar to that of the word "dog" we can conclude that there is statistical strength between the two events, which does not occur when using high-dimensional vectors in which each of the words will be associated to its own dimension and the occurrences of one word will not give us information about the occurrences of the other.

As previously mentioned, we can encode categorical data which is to be used by a classifier as one-hot encodings and dense embedded vectors. Categorical data refers to letters, words, sentences, part-of-speech tags and other linguistic features. Commonly, the structure of NLP classification system that is based on a neural network is the following (Goldberg, 2016):

a) Extracting a set of linguistic features f1,...,fk that are relevant for predicting the output class,
b) Retrieving the vector v(f1) for each feature fi,
c) Combining the vectors into an input vector x,
d) Feeding the vector x into a non-linear classifier (feed-forward neural network).

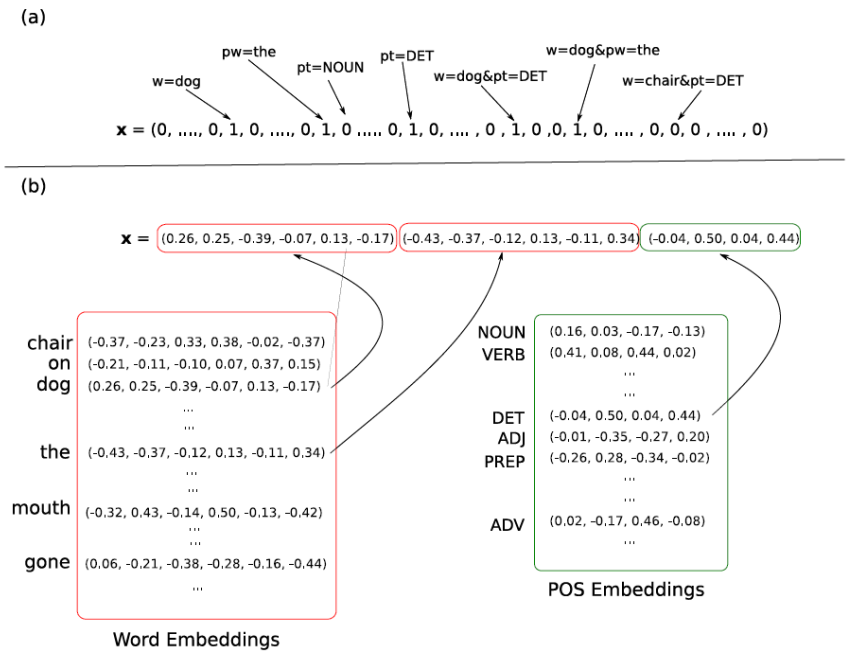


Image 1 - Sparse and dense feature representations (Goldberg, 2016)

The image, by (Goldberg, 2016) represents the distinction between a sparse feature vector (a) and dense, embeddings-based, feature vector (b).

In the sparse feature vector, each dimension represents a feature and feature combinations receive their own dimensions, which leads to the high dimensionality of the vector. When it comes to dense vectors, each core feature is represented as a vector and corresponds to several input vector entries which leads to low dimensionality. Feature to vector mappings are created on the basis of an embedding table (Goldberg, 2016).

The main distinction between one hot and dense vectors is in the dimensionality. When it comes to one hot encodings, each feature is in its own dimension and the dimensionality of the one-hot vector is the same as the number of distinct features (Goldberg, 2016). The features are independent from one another, e.g. the feature "word is 'cat" and "word is 'thinking'" are equally dissimilar to the feature "word is 'dog'".

In dense vectors, each feature is a d-dimensional vector and the dimensionality of a particular vector is d. Similar features have similar vectors, which means that some information is shared between similar features.

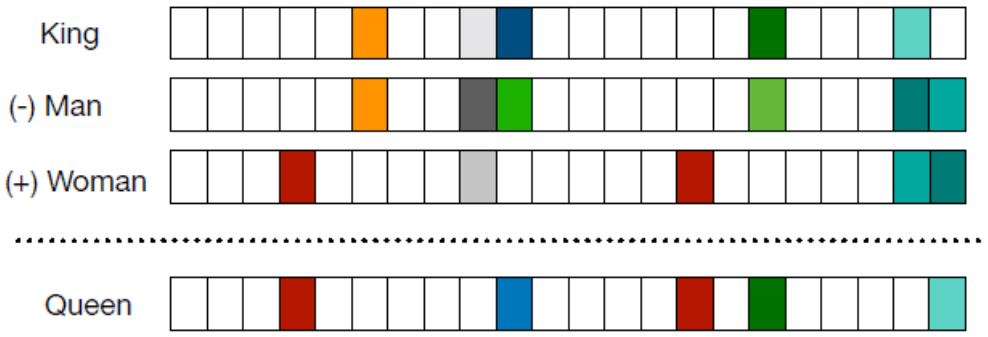


Image 2 - Distributional vectors represented by a D-dimensional vector where D<V and V is the size of the vocabulary (Young, Harazika, Poria, & Cambria, 2018)

Mikolov et al (2013b, 2013a) made a significant impact on NLP by proposing various tools which are widely used today, such as continuous bag-of-words and skip-gram models for constructing distributed vector representations. They altered the way in which we use word embeddings and suggested new approaches to constructing Word2Vec. These methods, CBOW (Continuous Bag Of Words) and Skip Gram both involve neural networks. The main distinction between the two is that the former allows predictions of the current word from the context, while the latter predicts surrounding words or the context given the current word.

CBOW model (Mikolov, Corrado, Chen, & Dean, 2013b) uses continuous distributed representation of the context, unlike standard bag-of-words models. The non-linear layer is removed and the projection layer is shared for all words and their vectors are averaged. Words in the history do not affect the projection; however four future words and four history words are used as the input where the training criterion is to classify the current word. The CBOW model is a neural network model that contains a hidden layer which has N neurons. The input is a one-hot vector of the context word that has V neurons. The output layer is the softmax probability over all the words in the vocabulary and the layers are connected by weight matrix (Young, Harazika, Poria, & Cambria, 2018).

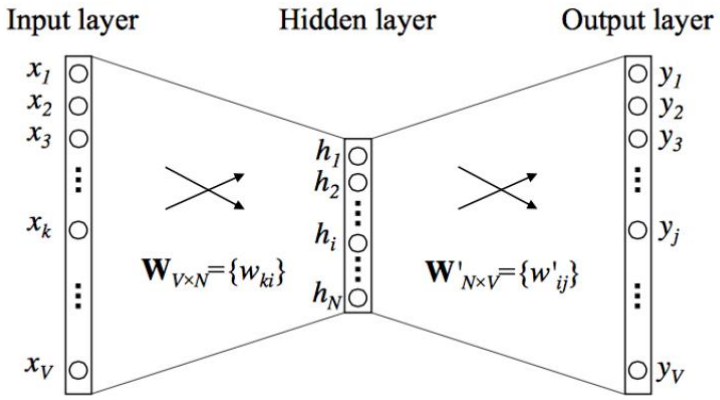


Image 3 - A CBOW model (Young, Harazika, Poria, & Cambria, 2018)

The skip-gram model tries to maximize the classification of a word based on another word in the same sentence or predicts words within a certain range before and after the current word. The resulting word vectors are achieved by giving less weight to distant words and increasing the computational complexity (Mikolov, Corrado, Chen, & Dean, 2013b).

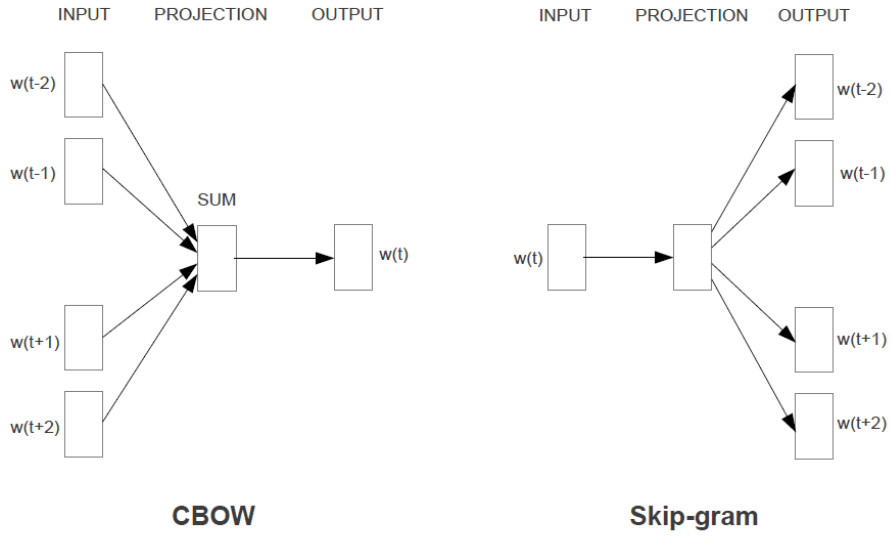


Image 4 - CBOW and skip-gram architectures (Mikolov, Corrado, Chen, & Dean, 2013b)

The way that word embeddings are designed was revolutionized by these models as they compute the conditional probability of a target word in relation to the context words surrounding

it. However, the limitations of this proposed model are based on the fact that if the window of surrounding words is small, sometimes contrasting words (such as “good” and “bad”) can share the same embedding, which creates issues when it comes to sentiment analysis or other NLP task regarding semantics. Furthermore, a problem arises with phrases and idioms combined of more words – such as “hot potato” (Young, Harazika, Poria, & Cambria, 2018).

GloVe (Global Vectors) is another word embedding method based on a count model where the word co-occurrence matrix is pre-processed by normalizing the counts and log-smoothing operation (Pennington, Socher, & Manning, 2014) (Young, Harazika, Poria, & Cambria, 2018). GloVe suggests representing each word as the sum of its corresponding word and context embedding vectors (Goldberg, 2017). It trains on global word-to-word co-occurrence counts and creates a word vector space (Pennington, Socher, & Manning, 2014).

Another type of word representation model was introduced recently by Peters et al (2018) called ELMo (Embeddings for Language Models). ELMo representations are vectors derived from a bidirectional LSTM that is pretrained with a language model on a large corpus of texts (Peters, 2018). The representations are contextual, deep and character-based. Contextual meaning that each representation is dependent on the context of the word and deep meaning that the word representations combine all layers of a pre-trained neural network (Allen Institute for Artificial Intelligence, 2018). Character-based means that the neural network uses morphological cues to form representations for out-of-vocabulary tokens (Allen Institute for Artificial Intelligence, 2018). ELMo proved to be successful when it comes to tackling various NLP tasks, among which are question-answering and sentiment analysis (Peters, 2018).

BERT (Bidirectional Encoder Representations from Transformers) (Devlin, 2018) is a language representation model which trains bidirectional representations from unlabeled text. It applies bidirectional training of the attention model-Transformer and gives a deeper look into language context by learning contextual relations between words in a text. The Transformer works by reading entire sequences of words at once, allowing the model to learn the context of the word based on the words that surround it (found on the left and the right side of the word). By adding an additional output layer it can produce state-of-the-art results when it comes to tasks such as question-answering (Horev, 2018). The first procedure in the process is applying a “masked LM (MLM)” in which 15% of tokens in a sequence are randomly masked which allows for the

prediction of the masked words, rather than constructing the entire input. The hidden vectors corresponding to the masked tokens are later fed into an output softmax. The second procedure is called Next Sentence Prediction and it allows the understanding of the connections between two sentences. This is achieved through pre-training for a binarized next sentence prediction task that can be generated from any monolingual corpus. BERT can be used for tasks such as sentiment analysis, sentence classification and Named-Entity Recognition (Devlin, 2018).

The FastText package, developed by Facebook Research, is a library which is used for learning text representations and text classifiers. FastText averages the word/n-gram embeddings to obtain sentence or document vectors. Furthermore, it uses multinominal logistic regression for the classification task. Joulin, Grave, Bojanowski and Mikolov (2016) presented TextRank and compared it with deep learning classifiers in terms of accuracy and speed. The testing ranked TextRank as being on par in terms of accuracy and in many cases faster than regular classifiers when it came to training and evaluation. The architecture averages word representations into text representation which are then fed to a linear classifier using multinomial logistic regression. They used a bag of n-grams to maintain the efficiency of the model without losing accuracy and a softmax layer to obtain probability distribution over pre-defined classes. The text representations can be shared among features and classes as a hidden state. In (Bojanowski, Grave, Joulin, & Mikolov, 2017), the authors use TextRank to learn word representations by taking into account subword information. The model outperformed baselines that do not take into account the context (the subword information) and the methods that are based on morphological analysis, while being fast and relinquishing the need for preprocessing and supervision. Furthermore, Joulin A. , et al. (2016) extended the FastText library with applying discriminative pruning in order to keep only relevant features of the trained model, preforming quantization of weight matrices and hashing of the dictionary with the purpose of reducing the complexity of text classifiers while maintaing accuracy and speed. FastText.zip proved to be faster, although not as accurate as the original FastText.

3.2.2. Convolutional neural networks

Convolutional neural networks have been highly successful in recent years, especially for computer vision tasks, while they can be useful in NLP tasks as well, such as summarization, sentiment analysis, machine translation and question-answering (Young, Harazika, Poria, & Cambria, 2018). Such neural networks produce vectors that are then used by other components of the network for preforming prediction tasks and can be used for feature extraction. CNNs identify n-grams without the need to pre-specify the embedding vector for each n-gram and thus allow sharing predictive features between multiple n-grams which share the same components (Goldberg, 2017). This architecture has proven to be successful for object detection and recognition of items that correspond to a predefined category regardless of their location in the image. The convolutions used in such computer vision tasks are 2D while when applied to text are mainly 1D convolutions (Goldberg, 2017). The pioneers in using CNNs for sentence modeling tasks are Collobert and Weston (2008), who considered six NLP tasks- Part-of-Speech tagging[2], chunking[3], name entity recognition (NER)[4], semantic role labeling (SRL)[5], language models[6] and semantically related words[7] (synonyms, homonyms, hyponyms…). They advocated for a deep neural network architecture which was trained end-to-end which included processing of the input sentence through several layers of feature extraction and automatically training the features by backpropagation in order for them to be relevant for each task. They used a look-up table layer in which each word was embedded into a vector in a d-dimensional space where the user defined the number of dimensions. By applying the lookup table to each word, sequences of words $\{s_1,s_2,\ldots s_n\}$ were then transformed into a series of vectors $\{v_1,v_2,\ldots v_n\}$ (Collobert & Weston, 2008).

---

[2] POS tagging aims at labeling each word with a tag that indicates its syntactic role (plural noun, adverb…)
[3] Chunking (shallow parsing) aims at labeling segments of a sentence with syntactic constituents (such as noun phrase NP)
[4] NER aims at labeling elements of a sentence into categories such as PERSON, LOCATION…
[5] SRL aims at giving a semantic role to a syntactic item in a sentence
[6] Language models estimate the probability of the next word in a sequence by labeling real texts and generating negative text
[7] SRL aims at predicting words which are semantically related, measured by the WordNet (http://wordnet.princeton.edu)

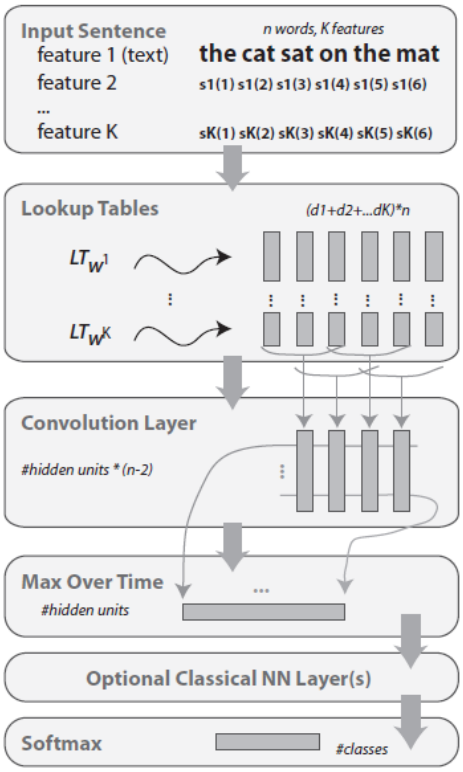


Image 5- NN architecture for NLP tasks (Collobert & Weston, 2008)

Convolutional neural networks can also be described as the process of convolution and pooling, in which a convolutional architecture can consist of multiple layers. The convolutional architecture identifies local aspects in a large structure, or n-grams that are considered to be predictive for a specific task. The most informative n-grams are then combined to create a vector representation of the structure; therefore it is not necessary to create a separate vector for each n-gram. When the convolutional architecture consists of more convolution layers, each layer can deal with a longer range of n-grams in the sentence (Goldberg, 2016).

The process of convolution and pooling begins by applying a non-linear, learned function (typically called a "filter") over each instantiation of a k-word sliding window over a sentence and transforming it into a scalar value. This process then results in a d-dimensional vector, in which each dimension corresponds to a filter (Goldberg, 2016).

The pooling process is performed by taking the maximum or the average value of each of the d-dimensional vectors and then combining the vectors into a single d dimensional vector. This results in finding the most relevant items in a sentence, regardless of their position. The created d–dimensional vector is then fed to the network in order to obtain predictions (Goldberg, 2017).

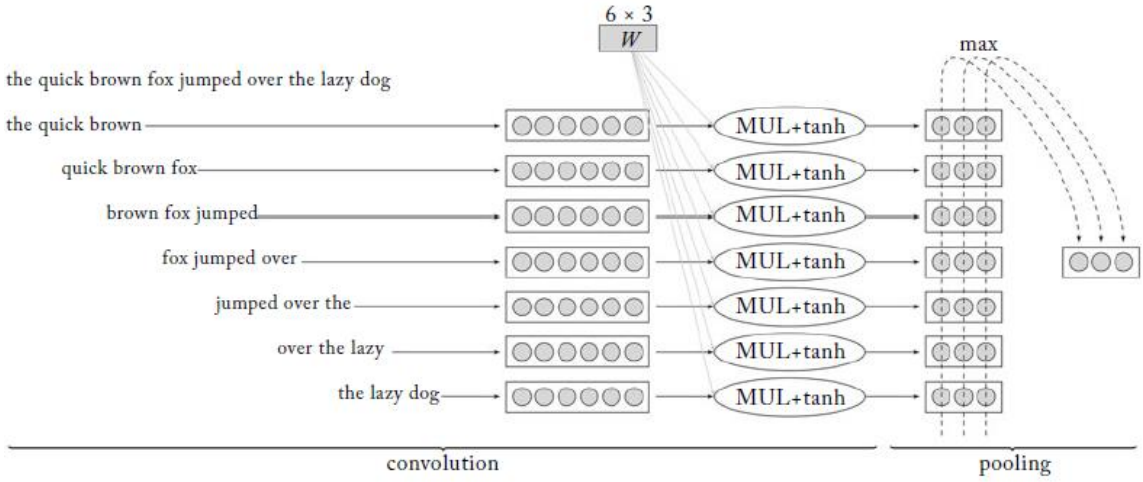


Image 6- The convolution and pooling process with a max pooling operation on a sentence (Goldberg, 2017)

The image (Goldberg, 2017) represents 1D convolutional and pooling process conducted over the sentence "the quick brown fox jumped over the lazy dog". In the example, each word was transformed into a two-dimensional vector, after which all the vectors were combined resulting in seven six-dimensional window representations. Those were then trained through a 6x3 filter which lowered the number of the dimensions and produced seven 3-dimensional representations. After that, a max-pooling operation was conducted, during which the maximum was taken over each dimension which then resulted in the final three-dimensional pooled vector.

The basic CNN can be used for sentence modeling as well as in the window approach (Young, Harazika, Poria, & Cambria, 2018). When it comes to sentence modeling, a sentence is represented as an embedding matrix, W∈ $R^{n\,x\,d}$ in which the words are represented as n and d is the dimension of the word embedding (Image 7).

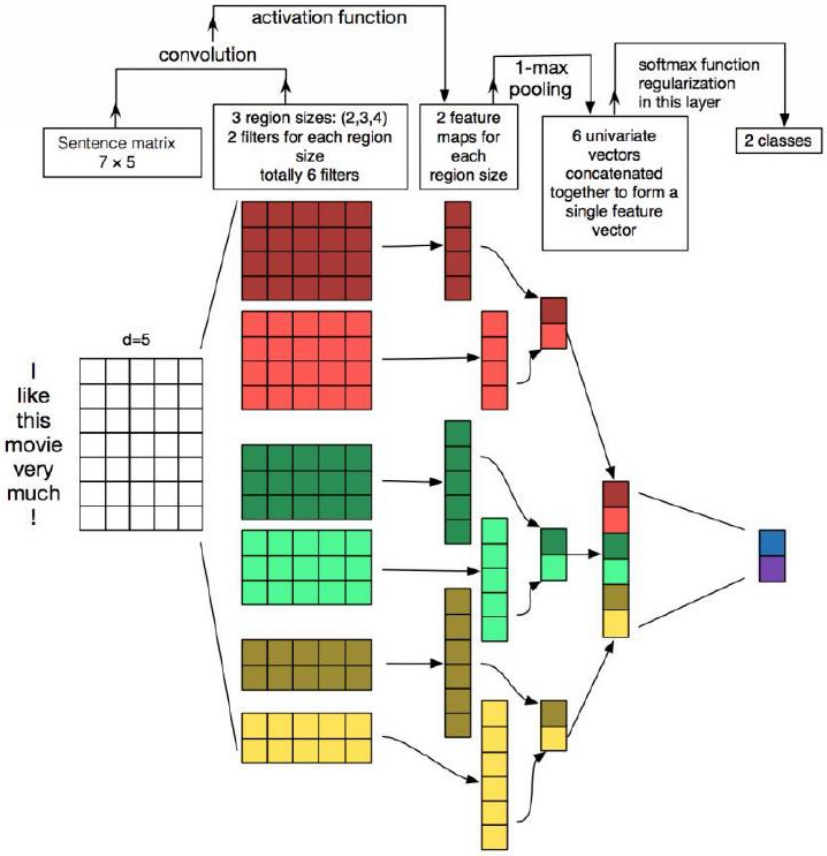


Image 7- CNN network modeling a text (Young, Harazika, Poria, & Cambria, 2018)

The convolutional filters also called kernels, slide over the entire word embedding matrix and extract a specific n-gram pattern. This process is followed by a max pooling process in which a max pooling operation is applied to each of the filters which results in a fixed-dimensional output. This is required for the process of classification and also it reduces the output's dimensionality by selecting only the most informative items of the sentence (Goldberg, 2016).

When it comes to the window approach, the predictions are based on words, as that is the demand of many NLP tasks, such as POS tagging, NER[8] and SRL[9]. This approach is based on the assumption that the role of each word in a sentence can be discovered when knowing the roles of the neighboring words. Therefore, the tag of one word depends on the tags of the words

[8] NER – Name Entity Recognition
[9] SRL – Semantic Role Labelling

that surround it. This is performed by attributing the predictions to the word in the center of the window and applying the CNN. The aim of this approach is to be able to obtain the tags of all the words in the sentence after the process is done (Goldberg, 2017).

3.2.3. Recurrent neural networks

RNNs are nowadays considered to be the most important deep learning approach when it comes to NLP tasks. Recurrent signifies that the network is able to take not only the input being given at the moment, but also what was given in the past – therefore recurrent means there is an order in time between the data (Goldberg, 2017). RNNs "remember" previous computations and use that information for preforming the current ones. Recurrent here also means that the network performs the same task over each instance of the sequence so that the output is a result of the previous computations (Young, Harazika, Poria, & Cambria, 2018). RNNs are used in the fields of automatic summarization, caption generation, speech recognition, machine translation and many other NLP tasks.

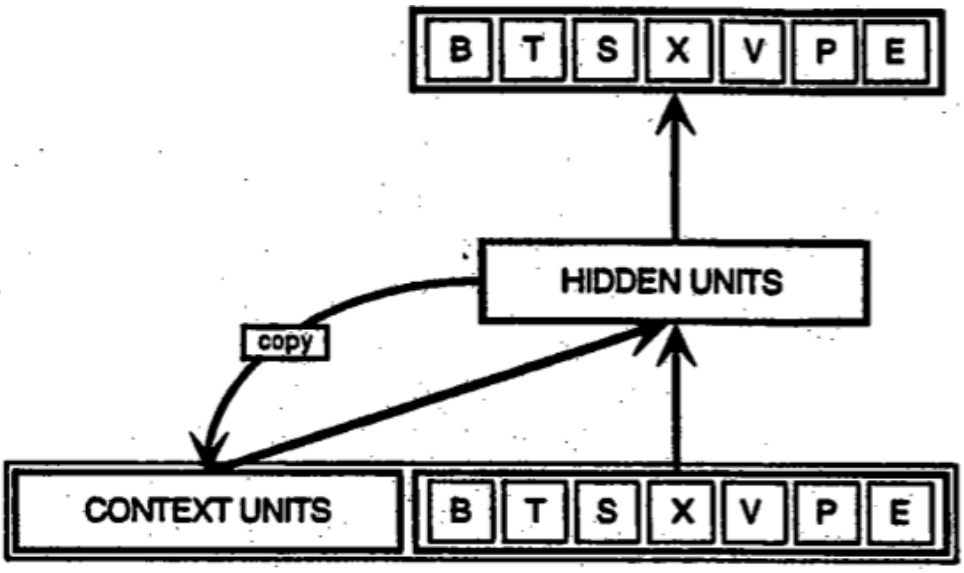
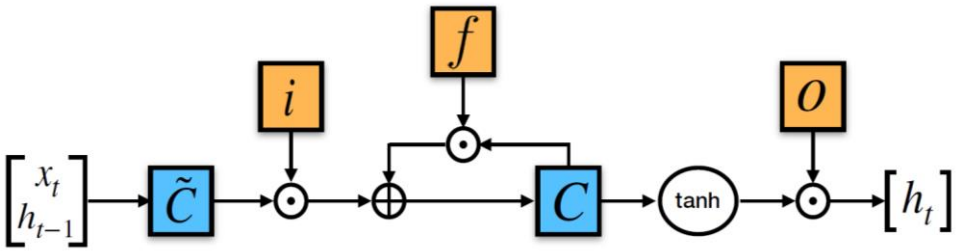


Image 8- a simple RNN architecture (McClelland, 2015)

Image 8 shows a simple recurrent network (SRN) that was first devised by Jeff Elman (Elman, 1990) which is a feed-forward back propagation network. The network has three layers and the input is an unbroken stream of letters from the alphabet (McClelland, 2015).The vital part of the architecture is the hidden state which can be considered as the memory of the network. However, with this type of RNNs a problem of the vanishing gradient can occur which sometimes prevents

the weights to change their value or even stops the network from continuing training. The vanishing gradient tends to occur when more layers using activation functions are added to the neural network, making the gradient of the loss function converge close to zero. If the gradient becomes too slow, issues with training the network arise. The weights of the initial layers are not updated, thus causing an inaccuracy over the whole network. (Wang, 2019). The second problem that can occur when using a simple RNN is the problem of exploding gradients. It occurs when error gradients accumulate during the update stage and produce large gradients and large values of weights. This creates an unstable network that can, in extreme cases, overflow and produce NaN results (Brownlee, 2017).

This problem has been solved by various advances in the architecture of networks, such as long short-term memory (LSTM) and gated recurrent units (GRUs) (Young, Harazika, Poria, & Cambria, 2018).



- **Image 9-LSTM gate (Young, Harazika, Poria, & Cambria, 2018)**

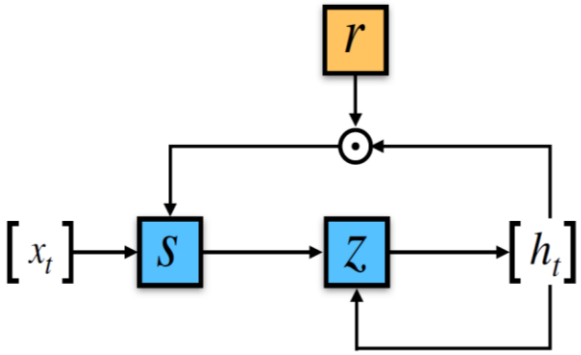
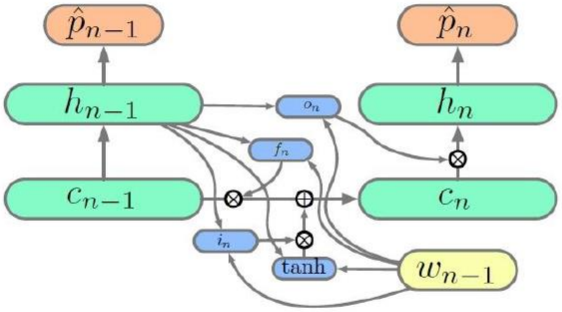


Image 10 GRU gate (Young, Harazika, Poria, & Cambria, 2018)

LSTM and GRU are the most used RNN architectures that differ from the simple RNN on the basis of additional gates. LSTM has additional “forget”, “input” and “output” gates, while GRU have two gates – a “reset” and an “update” gate (Goldberg, 2017).

LSTM (long short-term memory) is found to be very effective but quite complex as its forget gates are able to easily solve the vanishing gradient problem as well as the exploding gradient problem. It allows the error to back-propagate through an unlimited number of time steps. LSTM is comprised of input gates, forget gates and output gates (Young, Harazika, Poria, & Cambria, 2018). The purpose of an LSTM is to calculate the hidden state using a combination of the three gates. A vector (memory cells) is introduced that preserves gradients across time (Goldberg, 2017). The memory cells are controlled by the gates, while at each input state a gate is used to decide how much new input should be written to the memory cell and how much of the current content of the cell should be forgotten (Blunsom, 2017).

$$c_n = f_n \circ c_{n-1} + i_n \circ \tanh(V[w_{n-1}; h_{n-1}] + b_c)$$
$$h_n = o_n \circ \tanh(W_h c_n + b_h).$$



Hidden state

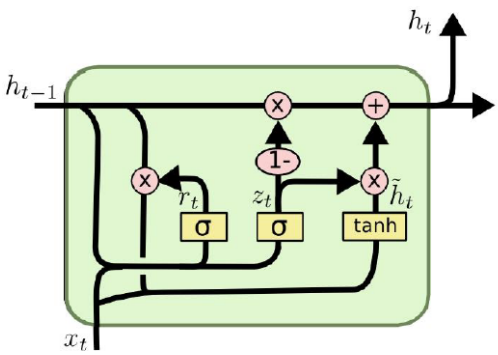
Input gate $i_n = \sigma(W_i[w_{n-1}; h_{t-1}] + b_i)$,

Forget gate $f_n = \sigma(W_f[w_{n-1}; h_{t-1}] + b_f)$,

Output gate $o_n = \sigma(W_o[w_{n-1}; h_{t-1}] + b_o)$.

Image 11- LSTM (Blunsom, 2017)

GRU (gated recurrent units) is a less complex solution than LSTM, although they might be the most successful RNN architecture at the moment (Goldberg, 2017). It consists of two gates – the reset gate and the update gate, while it lacks the separate memory component the LSTM has, but nonetheless secures the flow of information and prevents the emergence of gradient issues (Young, Harazika, Poria, & Cambria, 2018). The pictured gate r_n is used to control the access to the previous state h_{t-1} and to compute an update. The update (h_n) is determined based on the interpolation of the previous state and the proposal of the future state. The proportions of the interpolation are controlled using the gate z_n (Blunsom, 2017).



blend of old state and proposal state

$$h_n = (1 - z_n) \circ h_{n-1} + z_n \circ \hat{h}_n.$$

$$z_n = \sigma\left(W_z[x_n; h_{t-1}] + b_z\right),$$
$$r_n = \sigma\left(W_r[x_n; h_{t-1}] + b_r\right),$$
$$\hat{h}_n = \tanh\left(W_{\hat{h}}[x_n; r_n \circ h_{n-1}] + b_{\hat{h}}\right).$$

Image 12- GRU (Blunsom, 2017)

4. Evaluation methods

Evaluation is a crucial part of any NLP application. As new, improved summarization techniques are being developed, effective evaluation methods are crucial to measure (quantify/asses) the gap between the human-made summary and the automatic summary.

Evaluation measures can be divided into intrinsic and extrinsic, from which the intrinsic evaluation measures evaluate the outcome of the summarization and the extrinsic methods evaluate summaries based on the performance of the tasks the summaries were created for (Dong, 2018). When it comes to intrinsic evaluation, the automatically made summaries are compared with human-made "gold" standard summaries, which are also called the reference summaries.

The automatically created summary is of better quality if the distinction between the human made and automatically generated summary is less significant The intrinsic evaluation measures are easily replicated, however the difficulty arises when it comes to usefulness in the real world. There are many purposes for which a summary can be created and the summary should be evaluated in the context of that task or purpose. Some of the most popular extrinsic evaluation measures are relevance assessment (a rater decides if a summary is relevant to the topic or the event of the full text or transcript), reading comprehension (in which a rater is given the summary and multiple-choice questions about the text which will result in assessing the quality of the summary) and a decision audit task (in which users are presented with both the transcript and generated summary while their task is to write a synopsis of the most important information – raters evaluate the synopsis and a meeting browser inspects how often they clicked on the automatically generated summary) (Carenini, Murray, & Ng, 2011).

The problems which arise with all extrinsic evaluation methods are inter-rater and intra-rater reliability. Intra-rater reliability refers to the degree of agreement between multiple annotations of the text or multiple attempts by one rater to assess a summary. Inter-rater reliability is the degree of agreement between multiple raters or annotators. Due to a highly subjective task that is

creating and evaluating a summary, extrinsic evaluation can be impractical (Carenini, Murray, & Ng, 2011).

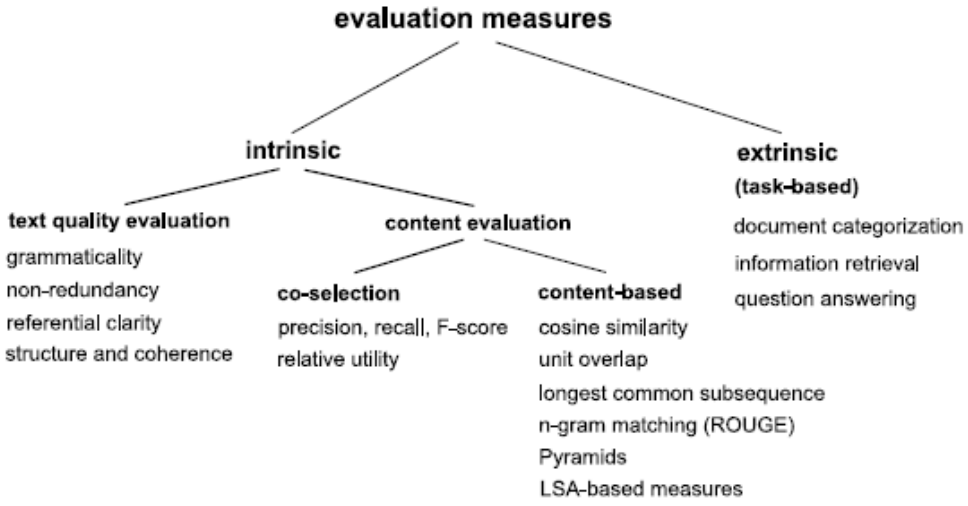


Image 13- Evaluation measures (Omar & Duru, 2017)

Important intrinsic metrics that can be classified as content evaluation and the most significant metrics of co-selection are precision, recall and F-score. Precision (P) refers to the number of sentences occurring in the system summary and the ideal summary, divided by the number of sentences in the system summary. Recall (R) refers to the number of sentences occurring in the system summary and the ideal summary, divided by the number of sentences in the ideal summary. Furthermore, F1-score acts as composition measure that unifies both precision and recall and is usually calculated by a harmonic mean of precision and recall (Steinberger & Ježek, 2009).

$$F1 = \frac{2*P*R}{P+R} \quad (3)$$

The most widely used intrinsic evaluation methods are ROUGE and Pyramid evaluation methods. Nowadays, a revised version of ROUGE (2.0) (Ganesan, 2018) is usually used for summarization tasks.

ROUGE (Recall Oriented Understudy for Gisting Evaluation) is a metric that consists of multiple evaluation methods that determine the quality of the automatically-made in comparison to the reference summary. Some of those evaluation methods that are usually used for the evaluation of automatic summarization are ROUGE-N (R-N), ROUGE-L (R-L) and ROUGE-SU (R-SU) (Dong, 2018).

ROUGE-N is a recall-based measure, which compares n-grams or series of n-grams (two, three and rarely four). It computes the percentage of the overlapping of the automatic summary and the reference summary. It uses consecutive matches of words in n-grams. The score is computed as:

$$\text{ROUGE-N} = n\ \frac{p}{q}, \tag{4}$$

where p is the number of common n-grams between the candidate and the reference summary and q is the number of n-grams from reference summary only (Allahyari, 2017).

Usually, for the task of evaluating summarization, ROUGE-1 (R-1) and ROUGE-2 (R-2) are used, which calculate the overlap of unigrams between the source text and the generated summary and the overlap of bigrams between the source and the summary, respectively.

ROUGE-L (R-L) is a measure that takes in account the longest common subsequence (LCS). The main gist is that it takes into account the sentence-level word orders and finds the longest in-sequence word overlapping. The longer the LCS is, the bigger the overlapping is and the more similar the two summaries are, which makes the candidate summary more successful and accurate (Allahyari, 2017).

ROUGE-SU (R-SU) measures the percentage of overlap of skip-bigrams (consisting of two words in a sentence with a gap between them) and unigrams. This allows the bigrams not to be only consecutive sequences of words, but allows for an insertion between the words (Dong, 2018).

The Pyramid metric (Nenkova, R, & McKeown, 2007) rates the summaries according to the semantic matching of the content units. For this metric, multiple reference summaries are necessary, because it works under the assumption that there is no perfect summary (Dong, 2018). The first stage of evaluation starts by the extraction of the most important information in the

reference summaries, which are called SCU or Summarization Content Units. These Content Units are usually no longer than a clause. The purpose of the metric is to compare multiple reference summaries with the purpose of detecting if they contain the same SCU. The sentences are indexed and annotated, followed by identifying similar sentences. Subsequently, the clauses are weighted according to the number of reference summaries they are found in. In Nenkova and Passonneau (2004) the sentences are indexed by a letter which represents the sentence the clause was extracted from and a number to represent the position of the sentence in the summary. After the sentences have been weighted, e.g.

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.

SCU1 (w=4): two Libyans were officially accused of the Lockerbie bombing

A1 [two Libyans] [indicted]

B1 [Two Libyans were indicted]

C1 [Two Libyans] [accused]

D2 [Two Libyan suspects were indicted]

SCU2 (w=3): the indictment of the two Lockerbie suspects was in 1991

$$\text{A1 [in 1991]}^2$$

$$\text{B1 [in 1991]}^2$$

$$\text{D1 [in 1991]}^2$$

(Nenkova, R, & McKeown, 2007).

The example above shows the identification of similar sentences after which the sentences are further compared to identify more tightly related subparts. Two SCUs are obtained from the underlined portions of the original sentences, after which they are weighted, first one having weight=4 and SCU2=3. The remaining parts of the sentence are contributors to the nine SCUs of different weights and granularity (Nenkova, R, & McKeown, 2007). The contributors express the semantic context of the sentence.

The SCUs are marked with a unique index, a weight and a natural language label (which serves to simplify the annotation process, to make sure that the annotator is conscious of the meaning shared by the contributors and a connection between the contributor and the context of the sentence). Nenkova and McKeown found 34-400 different SCUs in four 100-word summaries. As they increased the number of reference summaries, the number of SCUs grew.

After the entire document is annotated, the SCUs are partitioned in a pyramid, based on the weight of the SCUs, since each tier consists of SCUs with the same weight. Those with weight=4 are placed on top while those with weight=1 are on the bottom of the pyramid. The SCUs loose importance as the tiers descend since the ones that are on the bottom were located in fewer summaries. The optimal summary should consist of all the SCUs from the top tier and if the length of the summary allows it, some SCUs from lower tiers. The Pyramid scores express the proportion in which the content of the summary consists of the highly weighted SCUs. A summary can be considered of high quality if it contains the highly weighted SCUs with no or a small number of lower-tier SCUs.

The BLEU (Bilingual Evaluation Understudy) metric (Papineni, Roukos, Ward, & Zhu, 2002) is precision-based and it compares n-grams of the candidate summary with the n-grams of the source text. It measures how many words (or n-grams) in the machine generated summary appeared in the source document. If there are many words appearing in the summary that The BLEU metric ranges from 0 to 1 and multiple reference summaries can be provided at the same time. A recent addition to BLEU is the brevity penalty, which penalizes the system result if it is shorter than the length of the reference summary. However, BLEU is more often used with machine translation then summarization since BLEU does not consider the meaning or the

structure of a given sentence. Furthermore, it struggles with morphologically rich languages and it does not correlate closely to human judgement.

METEOR (Metric for Evaluation of Translation with Explicit Ordering) (Carnegie Mellon University, 2010) is a metric that is also most often used with machine translation and is based on the mean of unigram precision and recall. The limitations of this metric is that it does not account for fluency, grammaticality, coherence and other aspects of a quality summary, but rather relies mostly on lexical overlap, which creates issues when it comes to abstractive summarization. The metric was designed to be used with multiple reference summaries, to account for annotator subjectivity.

5. A survey on state-of-the art summarization techniques

In this section of the thesis, a survey on the state-of-the art summarization methods will be provided.

5.1. Extractive summarization

The task of extractive summarization of documents was tackled by Nallapati, Zhai and Zhou (2017) as they presented SummaRuNNer which is a sequence model for extractive summarization based on a Reccurent Neural Network and which was considered to produce state-of-the art results at that time. Aside from its performance, a big advantage of the model was considered to be his interpretability since it allowed for the visualization of predictions based on various features such as the content of the information, the novelty and the saliency.

They used a two-layer bidirectional GRU based RNN which has two gates (the update gate and the reset gate). The first layer of the RNN is considered to be bidirectional because it consists of two RNNs, from which the first one runs at the word level to perform the computation of hidden state representations at each word position, based on the current word embeddings and the previous state. The second one also performs at word level but it runs backwards from the last word to the first one. The second layer runs at sentence level and accepts the states of the first-layer RNN as input while the hidden states encode the representations of the sentences in the document. The representation of the document is modeled as a non- linear transformation of the average-pooling of the concatenated hidden states of the RNN and it can be considered as a summation of all the sentence-level hidden states where the weights are given by the probability of them appearing in the summary. When it comes to embeddings, 100-dimensional word2vec were trained on the CNN/DailyMail corpus.

The corpora used were the CNN/DailyMail corpus and the DUC single-document summarization dataset, while the performance of SummaRuNNer was evaluated using the Rouge metric while comparing the automatically made summary to the human-made, gold summary.

The results on the CNN/Daily Mail corpus demonstrated improvement of the state-of-the art results when it came to extractive summarization and were on par with the best results when it came to abstractive summarization. When it came to the Daily Mail corpus, the result was 26.6 on the Rouge-1 scale, 10.8 on the Rouge-2 scale and 14.4 on the Rouge-L scale for extractive summarization, while the results were 23.8 on Rouge-1, 9.6 on Rouge-2 and 13.3 on Rouge-L for the abstractive SummaRuNNer. On the joint CNN/Daily Mail corpus, the results were 37.5 on Rouge-1, 14.5 on Rouge-2 and 33.4 for Rouge-L with regards to abstractive summarization and 39.6, 16.2 and 35.3 respectively, for the extractive summarizer.

On the out-of-domain DUC 2002 corpus, SummaruNNer scored the following results: 44.8 on Rouge-1, 21.0 on Rouge-2 and 41.2 on Rouge-L for abstractive summarization and 46.6, 23.1 and 43.03 respectively, for abstractive summarization.

In a paper by Verma and Lee (2017) they combine some existing single-document summarization algorithms in one framework to be able to compare it to the then state-of-the art summarization tools. They donned the tool DocSUMM which revolves around TFIDF ranking and included both greedy and dynamic programming based algorithms. They proposed a "generalized" summarization model that unifies three dimensions of summarization (abstractive vs. extractive, single document vs. multi-document and syntactic vs. semantic). The results were on par with the state-of-the-art results at the time and the authors concluded that the then state-of-the-art tools only achieve about 54% of ROUGE-1 recall for single document summarization.

An extractive summarization technique was proposed by Mehta, Aurora and Majumder (2018) for summarizing scientific articles using pseudo-labeled data. They used a collection of scientific articles available through the ACL anthology as their dataset. In the experiment, 27801 articles were used, from which 23000 were used in the training stage, 2000 in the validation stage and 2801 as test sets. The authors proposed a context embedding technique with the purpose of determining the topic of a given paper using topic modeling and used a LSTM sequence encoder for learning attention weights across words. The summaries were evaluated using the ROUGE metrics. The model consisted of an LSTM based sentence encoder, topic modeling based context encoder, an attention module and a binary classifier. No manual labeling of the texts was preformed, but pseudo-labels were assigned for each sentence of the document based on their cosine similarity with the sentences found in the abstracts of the papers. The sentences were

labeled as 1 (important) or 0 (not important). The results on the ROUGE scale were 34.4 for R-1, 9.0 for R-2, 4.2 for R-3 and 2.7 for R-4.

The task of extractive summarization was also approached by Narayan et al (2017) who have taken into consideration the side information in the articles such as the titles and image captions to improve the results of automatic summarization. They conducted single-document summarization by using a neural network hierarchical document encoder and an attention based extractor over the side information. They donned the model SideNet. The model was tested on the CNN news highlights dataset and evaluated using the ROUGE scores. The hierarchical encoder decoder architecture consisted of RNNs and CNNs, while the CNN was used as a sentence encoder, one RNN as the document encoder and was used as an attention-based sentence extractor. The RNNs were in form of a single-layered LSTM architecture. The results on full length summaries are R-1: 54.2, R-2: 21.6, R-3: 12.0, R-4: 7.6 and R-L: 48.1. Human evaluation was also conducted and 20 random articles were selected from the dataset. The annotators were presented with the full-length articles and summaries from four different systems – SideNet, as well as two other models by various authors (LEAD baseline, PointerNeT) and the human authored highlights. The annotators' task was to rank the summaries from best (1^{st}) to worst (4^{th}) considering the informativeness and the fluency of the summaries. The human annotators ranked the human authored highlights as the best, while SideNet was placed second. The human authored highlights were ranked 1^{st} 48% of the time and SideNet was ranked first 28% of the time.

Furthermore, Narayan, Cohen and Lapata (2018) worked on using Reinforcement Learning to rank sentences for extractive summarization, as they envisioned extractive summarization to be merely a sentence ranking task. Additionally, they proposed a training algorithm which optimizes and rewards the ROUGE evaluation metric through a RL objective. Similar to the previous work by (Narayan, Cohen, & Lapata, 2018) and (Nallapati, Zhai, & Zhou, 2017) the summarization model consisted of a hierarchical document encoder (RNN with LSTM cells and a softmax layer) and sentence extractor (RNN with LSTM cells and a softmax layer. Reinforcement learning is used to optimize the metric used at test time, as the authors propose an objective function which combines maximum-likelihood cross-entropy loss with rewards from RL to optimize ROUGE. That makes the model better at distinguishing between sentences. The

authors named the model REFRESH (Reinforcement Learning-based Extractive Summarization) and tested it on the CNN corpus and the Daily Mail corpus as well as the two combined. Results were as follows: for the CNN dataset – R-1:30.4, R-2:11.7, R-L: 26.9; for the Daily Mail dataset: R-1:41.0, R-2:8.8, R-L: 37.7 and for the combined corpus R-1: 40.0, R-2: 18.2 and R-L: 36.6. The model outperformed models by (Nallapati, Zhai, & Zhou, 2017), (See, Liu, & Manning, 2017) and (Tan, Wan, & Xiao, 2017). The results show that the reinforcement learning allows for more fluent and coherent as well as informative summaries.

Wu and Hu (2018) also leveraged RL to improve on the coherence of generated extractive summaries. The authors combined a neural coherence model that can be trained end-to-end using unlabeled data and the ROUGE package which is used as a reward to create a RL method for the neural extractive summarizer. They named the tool RNES (Reinforced Neural Extractive Summarization) and its purpose was to learn to jointly improve coherence and informativeness of the summary. They used the CNN/ Daily Mail dataset to evaluate the model and the results for the framework trained with the coherence model were as follows: R-1: 40.95, R-2: 18.63 and R-L: 37.41. RNES trained without the coherence model achieved slightly better results, possibly due to ROUGE and the coherence objective undermining each other. However, human evaluation ranked the RNES with the coherence model more informative and coherent.

Tarnpradab, Liu and Hua (2017) preformed extractive summarization of forum threads. Summarizing forum threads can potentially be extremely useful for users who are searching for specific information in a forum or want to join the conversation but do not have the time to read a large number of posts. The authors introduced a supervised thread summarization approach, adapted from the neural hierarchical attention networks by (Yang, Yang, Dyer, He, Smola, & Hovy, 2016). The dataset consisted of 600 manually annotated forum threads with human summaries. The threads were taken from TripAdvisor and UbuntuForums. The authors used the ROUGE metric to evaluate their summaries and achieved the following results on the TripAdvisor dataset: R-1: 37.8, R-2:14.4 and R-1:37.6, R-2: 14.4 on UbuntuForums dataset. Furthermore, the sentence-level precision, recall and f-scores were calculated and the results were R- 32.5, P – 34.4 and F- 33.4 on TripAdvisor and R-33.9, P-33.8 and F- 33.8 on UbuntuForums. The results show that their model was able to capture the contextual information well and that the model preforms comparatively to other baselines.

The task of extractive summarization using neural networks was also approached by (Sinha, Yadav, & Gahlot, 2018) who proposed a data-driven approach using feedforward neural networks for single document summarization. The proposed model is based on a neural network which consists of one input layer, one hidden layer and an output layer. The document is fed to the input layer, the computations are done in the hidden layer and the output is generated at the final layer. The model was trained and evaluated on the DUC 2002 dataset and the ROUGE scores were R-1:55.1 and R-2: 22.6. The authors concluded that the model is fairly simpler in terms of implementation and memory complexity and that it can produce results comparable to sequence based models.

Zhou et al (2018) proposed a novel neural network framework (which they donned NEUSUM) for extractive summarization that unifies sentence scoring and sentence selection into a single step. The model simultaneously learns to score and select sentences by first reading the document sentences with a hierarchical encoder to obtain their representation and then generates the output summary by extracting the sentences one by one. The approach integrates sentence selection into the scoring model in order for the two tasks to benefit each other. Thus, sentence scoring can be aware of previously selected sentences and sentence selection can be simplified since the scoring function is learned to be the ROUGE score gain. The document is encoded in two levels –sentence encoding (bidirectional GRU RNN) and document encoding (BiGRU). In order for the model to be able to score the document sentences it should remember the information about the previous sentences and score the remaining document sentences based on the previously selected and the relevance of the remaining sentences. Another GRU is used as the recurrent unit to remember the partial output summary and MLP (Multi-Layer Perceptron) for sentence scoring. The authors conducted experiments on the CNN/Daily Mail dataset and evaluated it with the help of ROUGE. The scores were as follows: R-1: 41.59, R-2:19.01, R-L: 37.98. The results outperform (Nallapati, Zhai, & Zhou, 2017) and prove that the separation of scoring and selection might be beneficial for the task of summarization.

Zhang, Lapata, Wei and Zhou (2018) aimed to improve on the sentence-level labels when it comes to neural summarization by creating a latent variable extractive model. The main premise of the model is that the sentences are viewed as latent variables and the sentences that have activated variables are included in the summaries. The latent variable model views sentences as

binary values and used sentences with activated latent variables (ones) to infer gold summaries. The latent variables are predicted using the extractive model and the loss during training comes directly from the gold summaries. The model was evaluated on the CNN/Daily Mail corpus using the ROUGE evaluation metric and the results were : R-1: 41.05, R-2:18.77, R-L:37.54. The experimental results show that the latent variable model can improve an extractive summarization model.

Xie et al (2018) contributed previous research in the field of extractive summarization by improving and applying the WordNet based sentence ranking algorithm for extracting relevant sentences from a text. They claim that the algorithm helps with achieve better semantic relations between sentences in a summary. The method they used was based on the seq2seq attentional model such as in (Nallapati, Xiang, & Zhou, 2016). Two encoders based on the source text and extracted sentences were built in form of single-layer bidirectional Long Short-Term Memory (BiLSTM) and generating summaries achieved through a dual attention decoder (a single-layer unidirectional LSTM – UniLSTM). Out of vocabulary words and duplicate words are often an issue with automatic summaries and in this study it was addressed by combining pointer-generator and coverage mechanisms. The authors used the CNN/Daily Mail dataset and used 287112 pairs of articles and summaries for training. The preprocessing was done with the help of the Stanford CoreNLP toolkit. The authors used two methods for extracting sentences – *leading three* and modified sentence ranking algorithm based on WordNet. The summaries were evaluated with the ROUGE metric and the scores for the model combined with the *leading three* method are R-1: 39.41, R-2:17.30 and R-L: 35.92. The scores for the model combined with the WordNet sentence ranking algorithm are R-1: 39.32, R-2:17.15, R-L: 36.02. The results were on par with the other state-of-the-art ROUGE score, although this model had a fairly high R-2 score which demonstrates better readability of the generated summaries and higher quality semantic relations between the sentences.

Al-Sahabi, Zuping and Nadher (2018) also deal with the task of extractive summarization using representative learning through neural networks. They propose a model that addresses several issues of the previous modes, such as the memory problem and the issue of respecting the structure of the original document. Their model (HSSAS- Hierarchical Structured Self-Attentive

Model for Extractive Document Summarization) is based on a hierarchical structured self-attention mechanism to create the sentence and document embeddings. The neural-network based approach treats summarization as a classification task; as it calculates the score for each sentence taking into account the modeling features such as salience, redundancy and content richness. The model provides the following improvements – the hierarchical structure and the self-attention mechanism that is applied at word-level and at sentence level, which benefits the performance of the model and the selection of sentences. The authors used the CNN/Daily Mail and DUC-2002 datasets and the ROUGE metric for evaluation purposes. The results on the DUC 2002 corpus are as follows: R-1:52.1, R-2:24.5, R-L: 48.8. The results on the CNN/DM corpus were R-1:42.3, R-2: 17.8, R-L: 37.6.The proposed model outperformed models by (Nallapati, Zhai, & Zhou, 2017) and (See, Liu, & Manning, 2017).

Arumae and Liu (2018) developed question-focused rewards through converting human abstracts to a set of cloze-style comprehension questions and encouraging the system to extract salient source content which is useful for answering questions. The question body is a sentence of the abstract with a blank and the answer an entity or a keyword. The question is encoded into a vector using a bidirectional LSTM, which is then used to encode the summary into a sequence of vectors. An attention mechanism to find the segments of the summary that are relevant to the posed question so that the summary is able to answer multiple questions posed. The authors used RL for the task of extractive summarization and introduced a question-focused reward in order for the summaries to be more informative and relevant to the user. The results were compared with the results of the studies done by (See, Liu, & Manning, 2017), (Tan, Wan, & Xiao, 2017) and other earlier studies and have surpassed their results. The model was trained, validated and tested using the CNN dataset and the results on ROUGE were as follows: R-1:31.7, R-2:11.6, R-L: 21.5 for the keyword approach (which identifies the ROOT word of a sentence dependency parse tree and treats it as a keyword-based answer token).

In 2019, the field of automatic summarization continued to develop rapidly. Gehrmann, Layne and Dernoncourt (2019) have considered a broad application of ATS – using text summarization to aid readers with less developed reading skills. They recognized the effect of misleading headlines affecting readers' memory and reasoning skills, especially in the time of click-bait

when there are often discrepancies between the title and the actual content of an article. The authors aimed at developing a tool which would generate section titles that would provide a more concrete description of their topics. They created an extractive method that extract the most salient information by first selecting the most salient sentence and then applying deletion-based compression which is based on a Semi-Markov Conditional Random Field. It takes use of unsupervised word representations such as BERT and ELMo thus making the complex encoder-decoder architecture unnecessary. The selector was trained on the CNN/DM corpus and the compressor on the Google sentence compression dataset. The authors' evaluated the model on the full dataset using the ROUGE metric. The selector achieved the following scores: R-1: 30.2, R-2: 12.2 and R-L: 26.45. The authors also conducted a human evaluation in which 144 participants with different backgrounds (but who fluently spoke English) were provided with texts from various domains (Geography, Science, Anthropology, History), which either had a section title, a human-made title or no title at all, and the corresponding comprehension questions (six per text –advanced and intermediate level). In the total time of 30 minutes, the participants answered on average 68.25% of questions correctly and took about 16 minutes to complete the tasks when they were provided with section titles. In comparison, that was 30% faster than the fastest graduate student they recruited for the pilot testing. The approach performs as well as sequence-to-sequence models with unlimited training data, while it outperforms the same models in low-resource domains and the human evaluation proved that the section titles lead to improvements among multiple reading comprehension tasks.

Xu and Durrett (2019) reflected on the current methods of summarization which are either sentence-extractive or abstractive, using the seq2seq model to generate the summary. They present a neural model based on extraction and compression in which the model chooses sentences from the document and decides which compression method to apply on a certain sentence. For the extractive sentence selection, the authors use a bidirectional LSTM for encoding words after which multiple convolution and max pooling layers are applied to extract the sentence representations. After that, the sentences are aggregated into a document representation with a BiLSTM and CNN combination. The authors use a sequential LSTM decoder to produce a distribution over the remaining sentence representations (similarly to pointer-generator networks). When it comes to text compression; the authors use ELMo as a black box to compute contextualized word representations and CNN with max pooling to encode

the sentence. The concatenated representation is sent to a feedforward neural network to decide whether the compression span is to be deleted or to be kept, which is a classification issue. To tackle the issue of redundancy, the authors use the linguistically motivated compression rules and the parse tree for the model to compress the chunks with redundant information. The authors used the NYT and the CNN/DM corpora and evaluated the model using the ROUGE metric. The model achieved the following results on CNN/DM: R-1: 40.3, R-2:17.6, R-L: 36.4. The results on the NYT50 dataset were R-1: 44.3, R-2: 25.5 and R-L: 37.1. There is a significant difference between the performance on the two corpora, possibly due to CNN having shorter summaries overall, which points to compression showing its benefits more clearly on certain datasets. It surpassed models such as PointGen (See, Liu, & Manning, 2017) and NeuSum (Zhou, Yang, Wei, Huang, Zhou, & Zhao, 2018) on the CNN/DM corpus in terms of ROUGE.

Liu, Cheung and Louis (2019) abandoned the idea of content selection being an optimization problem as this type of approach ignores the usual structure of a human-written summary which can be exploited to train a summarization system. Their model, NEXTSUM, focuses on capturing the internal structure of a summary, based on the conviction that summaries in a certain domain often follow the same structure. NEXTSUM is an extractive summarization model that predicts the next sentence that should be included in the summary, using both the summary produced at that point (the output) and the source text. The model is comprised from a sentence prediction system and a summary generation model. The sentence prediction is a supervised system trained to select the next sentence from a set of candidates based on the source text and the summary created so far. The generation component builds a summary by making calls to the next-sentence predictor. The experiments were conducted on the NYT annotated corpus. NEXTSUM was evaluated using ROUGE-2 since the summaries are of different lengths and scores were assigned considering the domain of the article. For the domain of CRIME R-2 was 28.1, for ASSASSINATION the score was 24.1 and for the topic of BOMBS the score was 25.0. For summaries that pertain to mixed domains, the score was 24.1. The model outperformed other lead models and manages to produce similar summary lengths as those written by humans, based on the fact that people usually prefer summaries of different lengths depending on the topic of the query.

Liu (2019) leveraged BERT, which is a pre-trained transformer model (Devlin, 2018) that has performed well in many NLP tasks to create a single-document extractive summarization model named BERTSUM. To make using BERT for automatic summarization possible, the author modified the input sequence and embeddings of BERT, since BERT is trained as a masked-language model where the output vectors are tokens and not sentences. Liu uses interval sentence embeddings to distinguish multiple sentences within a document and applies an inter-sentence transformer rather than a sigmoid classifier on sentence representations. Furthermore, the author applies an LSTM layer over the BERT outputs, since combining RNNs with the Transformer model has proven to have advantages over the Transformer model on its own. The model was evaluated on CNN/DM corpus as well as the NYT dataset and rated with the ROUGE metric. It was compared to models by (Narayan, Cohen, & Lapata, 2018), (See, Liu, & Manning, 2017) and (Celikyilmaz, Bosselut, He, & Choi, 2018). The quantitative evaluation on BERTSUM+Transformer on CNN/DM produced the following results R-1: 43.25, R-2: 20.24, R-L: 39.63. All the models presented by the author outperform results achieved by the previously mentioned studies; however the BERTSUM with the LSTM lags after BERTSUM+ Transformer, but does not have an obvious influence on the performance of the model when compared to the model combined with a classifier. BERTSUM combined with the classifier produced the following results on the NYT corpus: R-1: 46.66, R-2:26.35 and R-L: 42.62. It outperformed models by Durrett et al (2016) and (Paulus, Xiong, & Socher, 2017).

BERT was used by Ga and Hu (2019) for the task of extractive summarization. They created a two-phase encoder-decoder architecture which is based on BERT and evaluated it using ROUGE and human annotators on the CNN/DM corpus. The structure of the model can be described in three steps – using the pre-trained BERT to fine-tune on CNN/DM, sentence classification and the production of the summary. Their model achieved the following results: R-1:37.30, R-2:17.05, R-L: 34.76. The human annotators were provided with samples from the CNN/DM dataset (input article, reference summary and output of the Bottom-up model (Gehermann, Deng, & Rush, 2018) and their own). They gave a total of 95/100 points based on relevance and readability to the authors' model. They rated the bottom-up model by (Gehermann, Deng, & Rush, 2018) with 80/100 points, thus the authors' model based on BERT achieved better performance than the bottom-up model when it came to the human annotators and lower results when it came to the automatic metric.

Miller (2019) also used a BERT model with the purpose of preforming extractive summarization. The author recognized the significance that summarization tools could have for education, especially when it comes to MOOCs[10]. Transcripts of videos that are used as educational material for online courses are available; however it is often the case that the most relevant information is difficult to spot. Due to the need for lecture summarization tools, Miller created a python-based RESTful service which uses the BERT model for text embeddings and K-means clustering for locating the most relevant sentences.

Khan, Qian and Naeem (Khan, Qian, & Naeem, 2019) used K-means clustering as well for the task of extractive summarization and combined it with a TFIDF model. The extractive summarization is performed by using K-Means clustering with TFIDF following the notion of finding the true K value by conducting the following approaches – the elbow and the silhouette method. Clustering refers to the unsupervised approach with which documents can be organized according to their class or domain and K-means clustering is a partitioning method used in data mining in which an algorithm segregates N number of documents into K number of clusters while the value of K is either specified by the user or by using heuristic methods to find the true K value. The algorithm functions in the following procedure – the document is read, after which cleaning or preprocessing is conducted. The sentences are weighted using the TFIDF Scores weight and then a method of finding the value of K is conducted (either elbow or silhouette). After that the high frequency clusters are selected and the summary of selected sentences is displayed. The authors used a dataset from kaggle.com which consisted of news headlines, the summary of the article and the article itself. The articles were taken from Hindu times, Indian times and Guardian. The authors used the BLEU metric for evaluating their results, and the cumulative score for the Elbow resulting summaries for Doc1 was 0.39, while the cumulative score for Silhouette resulting summaries for Doc1 was 0.42. The silhouette resulting summaries consistently achieve better BLEU scores for each document. The created summaries and the results of the statistical measures show that this method produces quality outputs, even though there are still issues regarding redundancy.

[10] MOOC- Massive Online Open Course

Liu, Titov and Lapata (2019) dealt with the task of extractive summarization by relying on linguistically motivated document representations to generate summaries. The model, named SUMO (Structured Summarization Model) induces a multi-root dependency tree through refining the structures predicted by previous iterations and in that way, predicts the output summary. They attempted single-document summarization through the dependency discourse tree in which each root node in the tree is a summary sentence and the subtrees are sentences which relate to the summary sentence based on their content. The baseline encoder of the model is based on the Transformer architecture as the baseline is comprised of a sentence-level Transformer and a document-level Transformer. The Transformers compute the representation of each word and the contextual representation of each sentence in the document. The architecture eliminates recurrence and unwanted repetitions through a self-attention mechanism which models relationships between all words in a sentence. SUMO decides if the sentence should be a part of the output summary and induces the structure of the document as a multi-root tree. The addition to the Transformer, unlike in the baseline model, is that it uses structured attention to model summary sentences represented by root nodes and iteratively refines the structures to infer more complex and higher-quality structures. The testing was conducted on the NYT dataset as well as the CNN/DM dataset and the highest results were achieved on the DM corpus (R-1:42.0, R-2: 19.1, R-L: 38.0) and the NYT corpus (R-1: 42.3, R-2:22.7, R-L: 38.6). Furthermore, a human evaluation was conducted and SUMO achieved 65.3 points on the QA-based evaluation when the summaries were based on the CNN/DM corpus and 57.3 when based on the NYT corpus, thus outperforming models by (Narayan, Cohen, & Lapata, Ranking Sentences for Extractive Summarization with Reinforcement Learning, 2018), (Celikyilmaz, Bosselut, He, & Choi, 2018) and (See, Liu, & Manning, 2017). The authors conclude that SUMO preformed comparatively to other state-of-the-art models and induces meaningful tree structures.

5.2. Abstractive summarization

Hua and Wang (2017) used a neural network model to summarize news stories and opinion articles and dealt with the setup of in-domain and out-of-domain data in order to achieve higher quality summaries. The news articles were used as the source domain while the opinion articles were the target domain and the data was taken from both domains, since it assures better performance of the model if the data from the target domain is scarce. An attention sequence-to sequence model with a pointer-generator mechanism was used as well as two LSTMs - a bidirectional recurrent neural network as the encoder and a uni-directional RNN as the decoder. The evaluation metrics used were ROUGE (ROUGE-2 (R-2) which measures bigram recall and ROUGE-L (R-L) which measures the longest common sub-sequence) and BLEU. The results have shown that the pre-training step improves the summary of articles from the domain of news, while the performance on the opinion articles is roughly the same, which can be related to the genre, since summaries of opinion articles tend to contain novel words which might not be found in the article itself. Furthermore, Hua and Wang studied the effects of domain-adaptation where opinion articles are the target domain and used a training set of opinion for in-domain and mix-domain training. The results show that the model trained for news is superior when it comes to generating summaries consistent of tokens which we can find in the original articles, while the model trained for opinion articles preforms better when it comes to generating novel words which are not found in the input. Their model was successful at selecting salient information for the summaries even when trained on out of domain data. The results achieved through pre-training and in-domain training when it came to summarizing news articles were 24.2 for R-2, 34.5 for R-L and 22.4 on BLEU, while the results for opinion articles were 19.9 for R-2, 31.8 for R-L and 14.22 for BLEU. The authors evaluated the effects that domain adaptation had on the summaries and established that information is transferrable across domains and that the model trained on out-of-domain data pays more attention to entities that can be categorized as PERSON and less to entities that could be categorized as ORGANIZATION, while it is reversed for in-domain trained model which points to opinion articles having more information relating to the category of PERSON and less information relating to ORGANIZATION than news articles. When it came to in-domain news to news training the attention distribution for the category of PERSON the result was 7.9%, while for out-of-domain training (news to opinion) it was 8.7%

and the mix-domain training (news and opinion to opinion articles) the result was 15.1%. When it comes to the category of ORGANIZATION, the results were 10.9%, 6.9% and 8.2%, respectively.

Paulus, Xioung and Socher (2017) introduced a neural network based abstractive summarization model that included an intra-attention model which aimed to combat the problem of the production of repetitive and incoherent phrases. The inter-temporal attention model is used as a part of the encoder to record previous attention weights for each of the input tokens and warrants that different parts of the input sequences are used. The intra-attention model is used as a part of the decoder and takes into account previously decoded sequences, i.e. the words generated by the decoder. For generating the tokens, the decoder uses either a token-generation softmax layer or a pointer mechanism. A switch function is used for deciding at each step which of these two methods should be used for generating tokens. Furthermore, a new training method was introduced, which merged reinforcement learning and supervised word prediction. The method used was the self-critical policy gradient training algorithm, which maximized a specific discrete metric. Furthermore, a mixed training objective function was used which ensured an increase of quality and readability of the generated summary, rather than simply optimizing for a specific metric such as ROUGE, which does not guarantee that the summary is appealing and understandable for a human reader. The datasets used were the CNN/Daily Mail corpus and the New York Times (NYT) dataset. The method that combined reinforcement learning and mixed-objective learning with the intra-attention mechanism produced the following results for the CNN/Daily Mail dataset – 39.87 on ROUGE-1, 15.82 on ROUGE-2, 36.90 on ROUGE-L. When it came to the NYT dataset, the quantitative results for the same methods were 42.92 for R-1 and 26.02 for R-2. Some of the conclusions of this study were that the intra-attention mechanism improves performance of the model when it comes to generating longer summaries, such as in the CNN/Daily Mail dataset, while it does not improve the results on shorter output sequences, such as with the NYT dataset. This model surpassed the state-of-the art models of the time which used the CNN/Daily Mail corpus for producing summaries, as well as the SummaRuNNer extractive model (Nallapati, Zhai, & Zhou, SummaRuNNer: A Recurrent Neural Network based Sequence Model for Extractive Summarization of Documents, 2017). Furthermore, the summaries were also evaluated by humans with the purpose of establishing their quality and readability. The human raters scored a hundred randomly selected examples from the CNN/Daily

Mail dataset. The raters had to look at the ground truth summary, the original article and the summaries generated by different models and rate them on a scale of 1 – 10. The human raters evaluated the summaries generated with the help of the model which combined reinforcement learning and the mixed training objective function as the best quality summaries and gave them an average grade of 7.04 on the scale of readability and 7.45 on the scale of relevance.

Nema, Khapra, Laha and Ravindran (2017) dealt with query-based summarization, which aims to represent the information that is relevant for a given query. Their model was based on the neural encode-attend-decode paradigm, which tended to suffer from repeating certain phrases so they added a query attention model which aims to focus on different aspects of the query at different time steps and a diversity based attention model which attempted to solve the issue of repeating phrases found in the output. The model, based on the neural encoder-attention-decoder paradigm uses an encoder RNN for the query and document, an attention mechanism for the query and the document and a decoder RNN. The RNNs are consistent of GRU architecture. The encoders for the query and the document read the query/document from left to right and compute a hidden representation for each time step, while the attention mechanism for the query is based on the decoder which produces and output word at each time step, focusing on various parts of the query. The attention mechanism for the document assigns weights for each word in the document and encodes the relevant information from the document and the query. The diversity based attention model treats successive context vectors as a sequence and an LSTM cell to compute a new state at each time step. The authors created a dataset which would be suitable for query based abstractive summarization from Debatapedia, which is an encyclopedia of for and against arguments and quotes on critical debate topics (Debatapedia, 2011). The corpus they put together consisted of 663 debates with 53 overlapping categories such as Politics and Crime. The output represented a triple which consisted of the query, the document name and the abstractive summary. The evaluation method used was ROUGE-1, ROUGE-2 and ROUGE-L. The soft LSTM based diversity model provided best results among the methods the authors used for the experiments and achieved 41.26 on R-1, 18.75 on R-2 and 40.43 on R-L. The novel part of this model was the diversification mechanism which ensured diverse context vectors at successive time steps and paid attention to words in the history in case they are needed when generating the summary.

Li, Lam, Bing and Wang (2017) dealt with the task of abstractive summarization and brought a new framework based on a sequence-to-sequence oriented encoder-decoder model with a deep recurrent generative decoder. The model used for the latent structure information which is implied in the summaries is a recurrent latent random model, while neural variational interference is used to address the intractable posterior interference for the recurrent latent variables. The latent structure information can be defined as common structures within a text such as "What", "What-happened", "Who-action-What" and by taking advantage of this information, the authors aim at improving the quality of the generated summaries. To address that matter, they introduce a new framework based on sequence-to-sequence oriented encoder-decoder model with a latent structure modeling component. They used Variational Auto-Encoders as a base for their generative framework and added historical dependencies on the latent variables. Furthermore, they introduce a deep recurrent generative decoder (DRGD) for latent structure modeling which is integrated in a unified decoding framework with the standard deterministic decoder. A neural network based framework is used to tackle the variational interference and generation for the recurrent generative decoder component, in the form of GRU architecture. The datasets used were Gigawords, which is an English sentence summarization dataset, DUC-2004 ad LCTS which is a large-scale Chinese short text dataset. The evaluation metric used was ROUGE (R-1, R-2, R-L and R-SU4). The results of DRGD on the Gigawords dataset were 36.27 on R-1, 17.57 on R-2 and 33.62 on R-L. The results on DUC-2004 dataset were 31.79 for R-1, 10.75 for R-2 ad 27.48 for R-L. The results on the Chinese LCTS dataset were 36.99 on R-1, 24.15 on R-2 and 34.21 on R-L.

A great contribution to abstractive summarization was achieved by (See, Liu, & Manning, 2017) who proposed a pointer-generator network which improves on the standard sequence-to-sequence attentional model. The network copies words from the source text through pointing and produces novel words with the help of a generator. Furthermore, coverage is used to take into consideration the already summarized parts of the text. In this way, the model aids with accurate and fluent representation of the text as well as tackles the issue of repetitive phrases with the help of the coverage mechanism. The pointer generator model manages to solve the issue of OOV words; however it still leaves the problem of repeating phrases, which is then solved with the help of the coverage mechanism. The tests were done on the CNN/Daily Mail dataset with the help of the ROUGE and METEOR metrics. The method achieved the following results: R-

1:39.53, R-2:17.28, R-L:36.38 and 17.32 on Meteor in exact match mode and 22.21 in full mode (which rewards matching stems, synonyms and paraphrases).The model surpassed the (Nallapati, Xiang, & Zhou, 2016) model however it did not surpass the extractive SummaRuNNer model by (Nallapati, Zhai, & Zhou, 2017). However, the pointer-generator framework was an advance on which further research was built.

Liao, Lebanoff and Liu (2018) dealt with the task of multi-document abstractive summarization by using Abstract Meaning Representation (AMR), which is a semantic representation of natural language based on linguistic theory. AMR aims to represent documents as sets of summary graphs which are later transformed into sets of summary sentences. The graphs are rooted, directed and acyclic, the nodes represent the concepts and the edges stand for semantic relations. One benefit of this method is that it is not domain-specific, which makes it fitting for the task of abstractive summarization. The framework consisted of three steps: content planning, surface realizations and source sentence selection. During content planning, each sentence is mapped into an ARM graph after what all the sentences are merged into a connected source graph, from which a summary graph is extracted using structured prediction. After that, in the phase of surface realizations, the summary graph is converted into its PENMAN representation. Furthermore, a natural language sentence is generated from the PENMAN representation and during the stage of source sentence selection; sets of similar sentences are extracted from the source documents, while taking into account the topic of the sentences. This is done through spectral clustering. The selected sentences are fed to the component assigned to content planning with the purpose of generating summary AMR graphs. The datasets used were DUC-2004 and TAC-2011. The results indicate that using the AMR formalism for multi-document summarization can be beneficial as the results are on par with state-of-the-art abstractive approaches when it comes to abstractive summarization.

Dohare, Gupta and Karnick (2018) also dealt with Semantic Abstractive Summarization by using the AMR graph technique which creates summary sentences from a summary graph which is generated using the co-reference resolution and Meta Nodes. The authors used the proxy report section of the AMR Bank because it contains human-generated AMR graphs for news articles as well as their summaries. One of the crucial steps in creating the story AMR was the node co-reference resolution by implementing multiple sanity checks to avoid wrong mergers. They used

alignments which provide a mapping from a word in the text to the corresponding node in AMR. They introduced Meta nodes, as a new set of nodes to overcome the issue that co-reference resolution has with words not reinforcing the importance of each other in cases when words are describing the same event implicitly. The results for the ROUGE metric were R-1:40.9, R-2:16.7, R-L: 29.5 which outperforms (Liao, Lebanoff, & Liu, 2018); however the dataset was significantly smaller than the one (Liao, Lebanoff, & Liu, 2018) used in their study.

Another study that extended previous work on abstractive summarization using AMR was done by Hardy and Vlachos (2018). The authors aim to improve on the shortcomings of AMR, such as ignoring the aspects of language (tense, grammatical number etc.) by presenting an approach to guide the NLG (natural language generation stage) in AMR based abstractive summarization by using the information from the source document. They achieve this by estimating the probability distribution of the side information and using it to guide a seq2seq model (based on (Luong, Pham, & Manning, 2015) for NLG. The results on the CNN/Daily Mail dataset achieved through the ROUGE and BLEU metrics show that the guided metric has better scores than the unguided. The best results on the guided model were achieved using the RIGA parser and are as follows: R-1: 42.3, R-2: 20.3 and R-L: 31.4. The authors that better results can be achieved through jointly training the guiding process with the AMR-based summarization process. Kodaira and Komachi (2018) also approached the task of abstractive text summarization in their paper "Abstractive Text Summarization in Three Bullet Points". They claimed that the reason why previous abstractive summarizers had issues with repeating phrases in their summaries is not taking into consideration information structure, since the standard datasets consist of summaries of varying length which creates problems with the cohesiveness of the summaries. They aimed to solve that issue by using a dataset which consists of only three bullet points. They used a neural-network abstractive summarization model based on the model of Nallapati et al (2016) (2017). They used Japanese articles and summaries from Livedoor News which were written by human editors and consist of exactly three sentences. The dataset consisted of 214120 pairs of articles and summaries. The summaries were decided into four types: parallel, parallel with enumeration, sequence and sequence with segmented sentences. Enumeration refers to the introduction of new items in a summary while segmented sentences are those that were extracted from a larger sentence in the original document. The metric used for evaluating the summaries was ROUGE

and the results indicate that the proposed model improves performance of the summary. It scored 49.48 on R-1, 29.15 on R-2 and 35.82 on R-L.

Azunre, et al. (2018) conducted an interesting study dealing with the abstractive summarization of tabular data. They used an abstractive method accompanied by a knowledge based semantic embedding. The purpose of their study was to include the descriptive text in the headers, columns or other supporting metadata into the summary by employing knowledge based semantic embedding to recommend a subject/type for each text segment. The recommendations are then accumulated into a collection of super types which are descriptive of the dataset. The authors chose an abstractive approach since it works on building internal semantic representation and producing subject tags, which may not be clearly visible by simply extracting the supporting data (headlines, metadata, tabular data). They named their framework DUKE (Dataset Understanding via Knowledge-based Embeddings) which aims at employing a pre-trained Knowledge Base semantic embedding for preforming type recommendation within a specified ontology. The Knowledge Based embedding generates a dataset2vec embedding, Along with that, methods such as word2vec and wiki2vec are used. Word2vec is used to calculate the distances between the words in the dataset and the set of types in the ontology by building a vector space which maps words to points in a space where the proximity between the words stands for their semantic similarity. Wiki2vec is a form of word2vec already trained on a corpus of Wikipedia KB documents. The method created by the authors can be described in three steps: the sets of types and an ontology are collected to use for abstractive summarization, after which the text data is extracted from the tabular dataset and embedded into a vector space with the purpose of measuring the distance to all the types in the ontology and finally, the distance vectors for every keyword are aggregated into a single vector of distances. The authors used four randomly selected CKAN datasets (Class Size 2016-2017, Annual Survey Questions, Liquor Store Product Price List Oct 2017 and Coalfile Report), four OpenML datasets (185 baseball, 196 autoMpg, 30 personae, 313 spectrometer datasets) and data.world datasets (US terrorist origins, Occupation Employment Growth, CAFOD activity file for Haiti and Queensland Gambling data). When it comes to the first two CKAN datasets, DUKE predicted the exact tags and for the next two the accuracy was medium (“wine region” being close to “wine” and “river”). When it comes to the OpenML datasets, DUKE predicted the exact tags for the first two and had medium accuracy when it came to the second two (“person” similar to “personality”).

DUKE achieved medium accuracy on the first two data.world datasets and high accuracy on the second two.

Celikyilmaz, Bosselut, He, and Choi (2018) improved on the task of abstractive summarization by presenting deep communicating agents as a part of the encoder-decoder architecture. The aim was to address the issues that arise while summarizing long documents, as the task of encoding is then divided across multiple collaborating agents, from which each is in charge of a subsection of the input text. The encoders are connected to a single decoder (a single-layer LSTM) which is trained end-to-end using reinforcement learning and deliver information to the decoder with a novel contextual agent attention which ensures that the information from the encoders is integrated at each decoding step. The multi-agent encoder framework is built in a way that each agent encodes the word sequences with two stacked encoders - a local encoder and contextual encoder (bi-directional LSTM). The deep communicating agents are trained with a mixed training objective that optimizes multiple losses, comprised of MLE – the baseline multi-agent model uses maximum likelihood training for sequence generation, semantic cohesion loss is included in order to ensure that the sentences are informative and non-repetitive, reinforcement learning loss directly optimizes discrete target evaluation metrics and mixed loss guarantees learning a better language model and better results on evaluation measures. The datasets used were CNN/Daily Mail and the NYT dataset and the evaluation metrics used were ROUGE-1, ROUGE-2 and ROUGE-L. The best results on R-1 and R-2 were achieved with MLE, semantic cohesion and reinforcement learning with 3 communicative agents – 41-69 and 19.47, respectively. However, 3-agent models are usually outperformed by 5-agent models when it comes to ROUGE-L (3-agent model: 37.92, 5-agent model: 38.21) and the model falls short when it comes to the best published RL baseline. The authors' best multi-agent model (with three agents) does produce good quality summaries and adds extra details to the summaries, which makes them more diverse and useful.

Gehermann, Deng and Rush (2018) proposed including a data-efficient content selector that works as a bottom-up attention step to their neural-network based method for summarization. They aimed at improving the quality of the content that is being selected to comprise the generated summary. The benefits of this approach are the higher fluency and readability of the generated summaries and the fact that they only require about 1000 sentences for training, which

makes the model easily adaptable for any domain of text. The bottom-up approach works by first choosing the selection mask for the source document and then constraining the neural model for this specific mask. In doing this, the decision of which phrases are included in the summary is made more easily and there is no sacrifice of fluency when it comes to the final output. The abstractive summarizer is modeled with an attentional sequence-to-sequence model and a copy mechanism is used to copy words from the source. When it comes to content selection, it was defined as a word-level extractive task and treated as a sequence tagging issue. The training data was generated by aligning the summaries to the document and a standard bidirectional LSTM trained with maximum likelihood was used for the sequence labeling issue. Each token is mapped into two embedding channels, one of which is a static channel of pre-trained word embeddings (such as GloVe) and the other contextual embeddings from a pre-trained language model (such as ELMo) which uses a character-aware token embedding followed by two bidirectional LSRM layers. The embeddings are concatenated into a single vector that is used as input for the bidirectional LSTM which then computes a representation for a certain word which allows for the calculation of the probability that that certain word is selected for the summary. The authors evaluated their approach on the CNN/Daily Mail corpus as well as on the NYT corpus which are standard for news summarization. The bottom-up attention model achieved 41.22 on ROUGE-1, 18.68 on R-2 and 38.34 on R-L when trained on the CNN-DM corpus. The model scored 47.98 on R-1, 31.23 on R-2 and 41.81 on R-L for NYT corpus. This shows that the combined bottom-up summarization system lead to improvements of ROUGE scores over two points on both corpora when compared to models by (Paulus, Xiong, & Socher, 2017) and (Celikyilmaz, Bosselut, He, & Choi, 2018), although the model requires fine-tuned inference restrictions. The model seems to be useful for data-efficiency and easy to transfer to another domain.

Al-Sabahi, Zuping and Kang (2018) dealt with the task of abstractive summarization and aimed to improve on sequence generative models with RNN variants by employing a bidirectional encoder-decoder model in which both the encoder and the decoder are bidirectional LSTMs instead of an unidirectional decoder. They attempt to solve the issue of the limits of unidirectional encoders when it comes to tackling long sequences of text. Additionally, they add a bidirectional beam search mechanism that stands as an inference algorithm for generating the output summaries from the bidirectional model, which enables the model to take into

consideration the past and future sequences which improves on the output. They adopt the pointer-generator network and coverage mechanisms from (See, Liu, & Manning, 2017), which tackles the issues of OOV (out-of-vocabulary words) and creates more novelty words and a higher quality of abstraction. The bidirectional encoder-decoder model is a bidirectional LSRM which consists of two layers – one of which learns the historical context and the other learns the future textual context. The output from the encoder is fed as input to the backward decoder while its output is fed to the forward decoder, after which a beam search mechanism is used to generate tokens for the final summary. The authors used the CNN/Daily Mail dataset to evaluate their model, as well as the ROUGE metric. The results they achieved (R-1: 42.6, R-2:18.8 and R-L: 38.5) outperform the SumaRuNNer (Nallapati, Zhai, & Zhou, 2017) as well as (Paulus, Xiong, & Socher, 2017) , (See, Liu, & Manning, 2017) and (Pasunuru & Bansal, 2018).

Song, Zhao and Liu (2018) also attempted to improve on the informativeness and relevance of abstractive summaries and maintaining the gist of the original text in an abstractive summary. They worked on structure-infused copy mechanisms that enable the tool to copy relevant words from the source sentences into the summary while still keeping the context intact. This approach combines source dependency structure with the copy mechanism of an abstractive sentence summarizer. They add on the popular sequence-to-sequence model which has previously shown good results by incorporating source syntactic structures in neural sentence summarization to make sure the individual summaries express the meaning of the source text. An example of that is maintaining the important parts of source syntactic structure such as dependency edge from the main verb to the subject to ensure that the issue of the "missing verb" does not occur in the generated summary. A two-layered stacked bi-directional LSTM is used as an encoder, to condense the entire text into a continuous vector and learn each representation for each unit (word, phrase) of the source text. They used an LSTM decoder with an attention mechanism to predict one word at a time. They implemented a copy mechanism such as in (See, Liu, & Manning, 2017) and added strategies to ensure that source syntactic structure is also included in the copy mechanism. The strategies were shallow combination – mapping structural labels to structural embeddings and 2-way combination (+word) which involves two attention matrices that represent the semantic aspect which is calculated as the strength of interaction between the encoder hidden state and the decoder hidden state. By merging the semantic and structural salience the authors were able to estimate how important a source word is to predicting an output

word. Furthermore, a strategy they used was 2-way combination (+relation) which takes into consideration the salient source relations such as between a subject and a verb, as they also play a significant role in word prediction. They perform this by capturing the saliency of the dependency edge pointing to a certain source word. They use a coverage-based regulizer which was proposed by (See, Liu, & Manning, 2017) and beam search with a reference mechanism. They evaluated their model on the Gigaword summarization dataset and the ROUGE evaluation metric. The results they obtained show that the model containing the structure-infused copy mechanism and the 2way+word strategy achieved the best results, namely R-1: 35.47, R-2:17.66 and R-L: 33.52, while the model with the 2way+relation strategy achieved better results only on R-1: 35.49. These results show that the model managed to outperform models by (Li, Lam, Bing, & Wang, 2017) and (Nallapati, Xiang, & Zhou, 2016). They also evaluated the linguistic quality of the summaries by hiring human raters who rated the summaries on the scale of 1 to 5 with regards to fluency, informativeness and faithfulness (to the original text). The model that used 2way+relation was graded as 3.0 on informativeness, 3.4 on fluency and 3.1 on faithfulness.

Pasunuru and Bansal (2018) achieved new state-of-the art results when tackling the task of abstractive summarization by taking into consideration logical entailment and non-redundancy of generated summaries. They used a reinforcement learning approach and introduced two novel reward functions – ROUGESal and Entail. ROUGESal weights up the salient phrases and words which are detected by the keyphrase classifier unlike the traditional ROUGE metric, which gives equal weights to every token. The Entail reward gives high scores to logically-entailed summaries with the help of an entailment classifier. The authors attempted to outpreform the pointer-coverage models by combining the novel rewards with a new multi-reward approach in which rewards are simultaneously optimized in aleternate mini-batches. The baseline of the model is a standard sequence-to-sequence single-layer bidirectional encoder and an unidirectional LSTM decoder with attention, pointer-copy and coverage mechanisms such as in (See, Liu, & Manning, 2017). They used the REINFORCE algortithm (Zaremba & Sutskever, 2015) to predict the next word and to update LSTM states and employ the SCST approach (Rennie, Marcheret, Mroueh, Ross, & Goel, 2016). The ROUGESal learns the saliency weights with the use of a saliency predictor which is trained on sentence an answer pairs from the SQuAD reading comprehension dataset. When the predictor is given a sentence, it assigns saliency probability to every token by using a bidirectional encoder with a softmax layer at every

time step of the hidden states. The Entailment reward is based on an entailment classifier which is trained on the SNLI and Multi-NLI datasets and it calculates the entailment probability score between the ground truth and each sentence of the generated summary. The authors tested their model on the CNN/Daily Mail dataset, as well as the DUC-2002 dataset doe testing and a combination of the SNLI and Multi-NLI corpora. The results on ROUGE are as follows: R-1:40.43, R-2:18.00, R-L:37.10 and they show that the model otpreforms models by (Nallapati, Xiang, & Zhou, 2016), (See, Liu, & Manning, 2017) and (Paulus, Xiong, & Socher, 2017). The authors also tested their models for saliency and according to their saliency prediction model, the results for the baseline model, the ROUGE(RL) model, the ROUGESal(RL) model and the Entail(RL) model and the results were 27.95%, 28%, 28.80% and 30.86% respectively, while when tested on the CNN/Daily Mail Cloze Q&A setup the results were 60.66%, 59.36%, 60.67% and 64.66%. They also tested the models for the level of abstraction by following the "novel n-gram overlap" approach (See, Liu, & Manning, 2017). Here the level of abstractiveness ig higher if there is a high number of novel n-grams in the generated summary. The authors conclude that their Entail model has the highest scores of abstractiveness (2-gram:2.63, 3-gram:6.56, 4-gram:10.26), while the ROUGESal model is also comparable in terms of abstractiveness (2-gram:2.37, 3-gram:6.00, 4-gram:9.50).

Guo, Pasunuru and Bansal (2018) continue their work on improving the accuracy and saliency of the information in the summaries as well as the logical entailment. They use multi-task learning with an auxiliary task of question generation and entailment generation. Question-generation teaches the model to find salient questioning-worthy information and entailment ensures that a summary is a directed-logical subset of the input document. The baseline pointer and coverage model is a sequence-attention-sequence model with a 2-layer bidirectional LSTM encoder and a 2-layer unidirectional LSTM decoder. The pointer mechanism and coverage mechanisms are constructed following (See, Liu, & Manning, 2017). The question generation task is set up with the help of the SQuAD dataset which contains question and answer pairs and the Entailment generation is set up according to (Pasunuru & Bansal, 2018). Multi-task learning is employed for the simultaneous training of the tasks of abstractive summarization, question generation and entailment generation. The authors used the CNN/Daily Mail corpus and the Gigaword corpus for the summarization task and the SNLI and SQuAD corpora for the entailment and question generation tasks. Among evaluation metrics, ROUGE and METEOR are used to obtain the

results. The multi-task model with improvements (question generation and entailment generation) achieved results which surpassed the state-of-the art results and are as follows: on the CNN/Daily Mail corpus R-1:39.81, R-2:17.64, R-L: 36.54 and METEOR: 18.54. The summarization results on the Gigaword corpus for the same model are R-1:35.98, R-2:17.76, R-L: 33.63. The human evaluation results also showed that the MLT model is better than state-of-the-art when it comes to relevance and readability.

Cohan, et al. (2018) proposed the first model for abstractive summarization of single documents which are longer in form, which can be applied for research papers. Most summarization models at this point were trained on the CNN/Daily Mail corpus which consists of shorter articles, which usually are comprised of around 600 words. This model is appropriate for significantly longer scientific papers and pays attention to the standard structure of such paper (methodology, experiments, results and conclusions). The authors' model is an abstractive model which includes a hierarchical encoder and a discourse-aware decoder. The hierarchical encoder consists of two single-layer bidirectional LSTMs, first of which encodes each discourse section and then the document, while the other focuses on the sections of the document consisting of sequences of tokens. The forward and backward LSTM states are combined to a single state by using a feed-forward network. A copy and coverage mechanisms are also included to avoid repetitions of phrases, according to (See, Liu, & Manning, 2017). The authors introduce datasets collected from scientific repositories – arXiv.org and PubMed.com. The results for the arXiv dataset are as follows: R-1: 35.80, R-2: 11.05, R-3: 3.62 and R-L: 31.80. The results on the PubMed dataset are R-1: 38.93, R-2: 15.37, R-3: 9.97 and R-L: 35.21. These results were proof that a neural sequence-to-sequence model can effectively summarize longer documents and was the basis for further improvement.

Kryściński, Paulus, Xiong and Socher (2018) proposed two new techniques for improving the level of abstraction of summaries, since the level of actual abstraction and the number of novel phrases tend to be low in automatically generated summaries. The authors aimed to improve the level of abstraction of the summaries by introducing two new extensions to the general model of summarization. They decomposed the decoder into a contextual network which aims to retrieve relevant information from the source text and a pre-trained language model that takes into consideration prior knowledge about language generation. The authors' second addition was a

mixed objective that optimizes the n-gram overlap with the ground-truth summary which betters the abstraction. Their summarization model follows (Paulus, Xiong, & Socher, 2017). The encoding is done via a bidirectional LSTM and the decoder uses temporal attention over the encoder sequence, penalizing the tokens that previously had high attention scores to prevent repetitions. An external language model is added to the decoder and it takes care of generating from the fixed vocabulary, while the decoder focuses on the attention and extraction, which allows for easily incorporating external knowledge regarding fluency and domain-specific language and style by pre-training the language model on large scale corpora. The language model is built as a 3-layer unidirectional LSTM with weight-dropped LSTM units. The authors created a new reinforcement learning abstractive reward by defining a novelty metric that encourages the production of novel words. They tested the model on the CNN/Daily Mail corpus and evaluated it with the ROUGE metric as well as by measuring the percentage of novel n-grams. The model that contained the external language model scored the best results on ROUGE: R-1: 40.72, R-2: 15.95, R-L: 38.14. However, the model that did not contain the external language model produced more novel bigrams (NN-1.: 3.19, NN-2: 22.79, NN-3: 39.9, NN-4:50.61) but had a lower ROUGE-2 score. The model accomplished state-of-the art results when it comes to R-L scores while their R-1 and R-2 scores were comparable with the state-of-the-art, while it outperformed previous abstractive approaches, which was measured by their abstraction metric.

Zhang, Tan and Wan (2018) intended to improve on multi-document abstractive summarization, since the neural summarization methods of the time had achieved great results when it came to single documents and short texts. The authors adapt the common seq-2-seq models for SDS (single-document summarization) to create a neural abstractive model for MDS (multi-document summarization). Their approach adapted a neural model which was previously trained on SDS to the MDS task and leveraged MDS training data to improve the pre-trained model. Their model is based on (Tan, Wan, & Xiao, 2017) SinABS. They adapted the SinABS model by adding a document set encoder to encode a set of document representation vectors into a document set representation, which adds another level to the hierarchical encoder structure and thus the encoder consists of three layers. Furthermore, due to the lack of dependency relationship between the documents, the authors do not use an LSTM as the document set encoder, rather weights are added to each document depending on the document itself and its contribution to the

document set. The decoder is a two-level hierarchical framework similar to SinABS. TextRank is used for computing the attention distribution, however unlike in SinABS, the attention is computed on all the sentences in a document set. The problem oh the amount of sentences was solved by setting up a more concentrated attention distribution and by allowing only the top K ranked sentences to have attention weights during the computation stage. The model was tested on the DUC-2004 corpus and evaluated with ROUGE. The model achieved the following results: R-1: 36.7, R-2: 7.83, R-SU: 12.4. To test if the model produces summaries with a satisfactory level of abstraction, they computed the edit-distance between each generated sentenced and the most similar original sentence to prove that the relevant fragments were not only copied verbatim. The ED on DUC-2004 was 22 and the word edit-distance for each generated sentence was 1.10 which verified that the generated sentences were indeed different from the original, which proved a level of abstraction. Human evaluators rated the linguistic quality of the summaries on three dimensions – coherence, non-redundancy and readability and rated them on a scale from 1 to 5. The results were coherence: 3.76, non-redundancy: 3.92 and readability: 4.08. The results of this study point to a promising success of this approach when it comes to MDS.

Baumel, Eyal and Elhadad (2018) similar to (Nema, Khapra, Laha, & Ravindran, 2017) dealt with query focused summarization and aimed to produce text with high coherence using abstractive methods. To make the current abstractive methods of text summarization applicable for the task of query-focused summarization, the authors incorporated query relevance in a pre-trained abstractive model, designed an iterated method to embed multi-document abstractive models within the multi-document requirement and adapted the target size of the generated summaries to a given ratio. The authors used the pointer-generator method by (See, Liu, & Manning, 2017) and modified it by including query relevance. The QFS task is divided into two parts: the first stage in which a relevance model determines the extent to which parts of the original text are relevant for the input query and the second stage in which a summarization method is applied to combine all the relevant information for the query in a coherent summary. The model was named RSA QFS (Relevance Sensitive Abstractive QFS) and evaluated using the DUC 2005, 2006 and 2007 datasets as well as the ROUGE metric. The iterative RSA Word count (which measures the simple overlap between the number of words in the query versus the number of words in the source sentence) achieved optimal results – R-1: 39.82, R-2:6.98, R-

L:15.73 for DUC2005, R-1:42.89, R-2:8.73, R-L:17.75 for DUC2006 and R-1:43.92, R-2:10.13, R-L:18.54 for DUC2007. The results outperform various extractive methods for query-based summarization; however there is steel a need to improve the coherence of the summaries.

Li, Bing and Lam (2018) introduced a training framework for neural abstarctive summarization which is based on the reinforcement learning actor-critic approaches. They sought to improve on previous methods that aimed to maxime the likelihood of the predicted summaries which lead to low-quality outputs with many incoherent sentences. The authors used a common seq2seq framework as the policy network for the actor. They used GRU as the cell for the RNN in the seq2seq model. As for the critic they combined a maximum-likelihood estimator with a global summary quality estimator which consists of a neural network based classifier. The policy gradient method was conducted to preform parameter learning and employed REINFORCE and Gumbel-Softmax to update the policy parameters of the actor. They used three popular datasets to evaluate their model: Gigawords, DUC-2004 and LCSTS (a Chinese short text summarization dataset). The evaluation mehod used was ROUGE and the results on Gigawords were as follows: R-1: 36.05, R-2: 17.35, R-L: 33.49. The results on DUC-2004 are R-1: 32.03, R-2: 10.99, R-L: 27.86 and the results on LCSTS are R-1: 37.51, R-2: 24.68, R-L: 35.02. The results show that the model has comparable results to those by (Nallapati, Xiang, & Zhou, 2016) and (Li, Lam, Bing, & Wang, 2017).

Amplayo, Lim and Hwang (2018) proposed Entity2Topic (E2T) which is a model that can easily be added to a sequence-to-sequence model based on an off-the-shelf entity linking system (ELS). Their aim was to improve on the functioning of the decoder in order for it to produce morecoherent and concise summaries with using linked entities which an be found in the original texts and together create the topic of the summary. The E2T module encodes entities extracted from the original text by an ELS and constructs a vector representing the topic of the summary to be generated as well as informs the decoder about the constructed vector. As the base model, the authors use a basic encoder-decoder RNN with a biGRU as the recurrent unit of the encoder and a two-layer uni-directional GRU for the decoder. When it comes to datasets, the authors used Gigawords and the CNN dataset and rated their model with the help of ROUGE evaluation metric. The results for the base model with E2T on Gigawords are as follows: R-1:37.04, R-2:16.66 and R-L:34.93 and the results on the CNN dataset are R-1:31.9, R-2:10.1 and R-L:23.9.

The results preform comparatively with other, more compley summarization methods such as (Tan, Wan, & Xiao, 2017).

Chen and Bansal (2018) designed a hybrid extractive-abstractive summarization model with policy-based reinforcement learning that selects salient sentences and rewrites them abstractively to generate a concise and coherent summary. In order to maintain language fluency in the summary, the authors created a sentence-level policy gradient method that bridges the non-differentiable computation between the two neural networks in a hierarchical manner. The model uses an extractor agent to extract the most salient sentences and then the abstractor network rewrites and paraphrases the sentences. An actor-critic policy gradient with sentence-level metric rewards is used to connect the two neural networks and to enable sentence saliency learning. The sentence-level reinforcement learning takes into account the word-sentence hierarchy, while the extractor combines RL and pointer networks. The abstractor is built as an encoder-alligner-decoder model with the copy mechanism and was trained on pseudo document-summary sentence pairs. The method brought a novel approach to summarization in which the advantages of the abstractive paradigm (concise rewriting of sentences and novel words generation) and the advantages of the extractive paradigm are joined together to improve on the speed and the quality of the model, thus creating a high-quality output. Usually, common models encode and attend to ever word in a long input document sequentially, however this model adopts a coarse-to-fine approach that simultaneously extracts and decodes sentences which leads to the lack of redundancy issues as the model firstly chooses the non-redundan sentences to astractively summarize. The model was evaluated with the standard ROUGE metric as well as the METEOR metric and the CNN/Daily Mail dataset was used for summarization. The results of the model (rnn-ext+abs+RL+rerank) on the original CNN/Daily Mail dataset are as follows: R-1: 40.88, R-2:17.80 and R-L: 38.54 while the result on METEOR was improved by removing the repetition-avoiding reranking strategy which removes the across-sentence repetitions. The reslut on METEOR with the rerank strategy is 20.38, while without it is 21.00. The model outpreformed models by (Nallapati, Zhai, & Zhou, 2017) and (Paulus, Xiong, & Socher, 2017) and the results were the new state-of-the-art at that point.

Keneshloo, Ramakrishnan and Reddy (2018) proposed a reinforcement learning framework that is based on a self-critic policy gradient approach. Using this approach, they attempted to enable

training on smaller text datasets as well as achieve good generalisation. Furthermore, the authors investigated if transfer learning could alleviate the problem of models being vunerable in generalizing to other datasets. Transfer learning was shown to be a good option for situation in which there is little or no ground-truth summaries. The authors proposed two tranfer learning models: the first one works by transferring network layers from a pre-trained model and the other which consists of a reinforcement learning framework which is given insights fom the self-critic policy strategy and produces a systematic mechanism that creates a trade-off between the avout of reliance on the source or target dataset. The authors used the pointer-generator model as basis for their RL framework, because of its ability to successfully tackle OOV (out-of-vocabulary) words. The RL framework has the addition of a self-critic policy gradient model, which ensures that the model focuses on the sequences that are superior than the greedy selection during training while at the same time, punishing those which are inferior. The authors tested the model on CNN/DM, Newsroom. DUC2003 and DUC2004 and evaluated the summaries using the ROUGE metric. The weighted average score on ROUGE was R-1:36.21, R-2:22.25, R-L:32.81. The authors demonstrated that the model is able to generalize on unseen test datasets and achieves state-of-the-art results with the help of transfer learning. The method they proposed has been able to achieve good results on various datasets, regardless of the size of the dataset.

During 2019, many significant improvements and interesting studies have been conducted in the field of abstractive text summarization. One of those is a paper by Ouyang, Song and McKeown (2019) who constructed a summarization system for cross-lingual summarization as well as summarization corpora for low-resource languages (Swahili, Somali and Tagalog). They used machine translation and the NYT corpus to automatically translate into English, and paired the noisy English input documents with clean English reference summaries to train the model, which provided better results than the current state-of-the art copy-attention abstractive summarizers on real-world Somali, Swahili and Tagalog documents. They trained three language-specific summarizers and evaluated the summaries on the documents originally written in the source languages as well as on a fourth language – Arabic. The evaluation done on Arabic documents pointed to the robust abstractive summarizers generalizing to unseen languages. The authors translated articles from the NYT corpus into each of the three low-resource languages using neural machine translation and translated them back into noisy English. They paired them with a clean English reference summary corresponding to the clean English article that generated it. In

that way, the abstractive model learns how to take an English input document with translation errors and to produce a fluent English summary. The authors implemented the (See, Liu, & Manning, 2017) pointer-generator network for their summarizers and pre-trained them on the unmodified NYT corpus. The performance of Somali NYT was evaluated with the ROUGE metric and the results were as follows: R-1: 38.07, R-2:15.76, R-L: 26.82. The performance of Swahili NYT was R-1: 39.96, R-2:17.56, R-L: 30.24 and the Tagalog NYT: R-1:40.96, R-2:18.91, R-L: 31.14. The results point to Somali being the most difficult language to attend to, but all three systems performed better than the baseline and produced more fluent English across source languages. They evaluated the system on Arabic as well, with the DUC 2004 dataset which consists of real-world Arabic news articles that are translated in English and paired with human-written summaries. The summarizers performed well even in comparison with the DUC 2004 systems on high-quality, human translated documents. The results: R-1: 29.43, R-2: 7 .02, R-L: 19.98, demonstrate that the summarizers have the ability to generalize and improve on the NYT baseline by improving the fluency of input documents which were automatically translated from a previously unseen language. The results point to the method being able to produce fluent summaries out of non-fluent inputs.

Another interesting study focused on the abstractive summarization of Reddit discussion posts (Kim, Kim, & Kim, 2019) from the TIFU subreddit, which is largely conversational in style. This study is novel regarding the addressed summarization task, because a great deal of users would largely benefit from having a forum summarized when wishing to join the discussion. The authors add that it is a common occurrence that the abstractive summarizers do not produce truly abstractive summaries, because they are trained to learn structural patterns, which creates bias. The authors aim to alleviate this bias by using a novel dataset for summarization- Reddit TIFU. Furthermore, they contribute the field by proposing a novel memory network model (MMN – multi-modal memory networks) which stores the information of source text from different levels of abstraction (word, sentence, paragraph and document-level). They claim that the MMN has an advantage over the seq2seq based models, because the model explicitly captures long-term information and builds representation of not only multiple levels but multiple ranges (e.g. sentences vs. paragraphs). The authors tested their model as well as other models such as PG (See, Liu, & Manning, 2017), SEASS (Zhou, Yang, Wei, Huang, Zhou, & Zhao, 2018) and DRGD (Li, Lam, Bing, & Wang, 2017) on the TIFU dataset. The MMN model achieved the best

results both in the long and the short dataset. The results for the long dataset are as follows: R-1: 19.0, R-2: 3.7 and R-L: 15.1. This study opened the gates for many further studies when it comes to the automatic summarization of informal online text, which in itself is a highly demanding task, due to the nature of the language used.

Karn, Chen, Chen, Waltinger and Schütze (2019) dealt with the summarization of multi-participant, threaded posting, such as comments on an article or a post, group chats or emails and discussions on a forum. Those types of conversation can be jointly called interleaved texts because several topics may take place concurrently. This creates issues for the task of summarization as the summarizer needs to recognize which sentences belong to which topic. The authors' model is an end-to-end encoder-decoder network which encodes interleaved posts hierarchically (word-to-word followed by post-to-post) and the decoder generates the summaries hierarchically (thread representation first followed by generating summary words). Furthermore, the authors proposed a novel hierarchical attention mechanism that is integrated in the encoder-decoder architecture. The encoder was built based on (Nallapati, Zhai, & Zhou, 2017) and for the decoder, the authors used threadLSTM in which the top level is a thread-to thread decoder (a unidirectional LSTM) and the low level a word-to-word decoder (a unidirectional attentional LSTM). The results on ROUGE were R-1:41.76, R-2:16.89, R-L:30.70 for the model that included hierarchical attention on the decoder.

Gao et al (2019) collected document-summary-comment pair data from Weibo (Chinese social media site) to create abstractive summaries which included users' comments. To tackle the issues of the informal and noisy nature of the comments as well as the fact that jointly modeling the news documents and the reader comments is challenging, as they differ in style, the authors create an adversarial learning model – RASG (reader aware summary generator). The model consists of a seq2seq summary generator, a reader attention module, a supervisor modeling the semantic gap between the summary and the reader focused aspects and a goal tracker that produces the goal for each generation step. The model proved its effectiveness by outperforming models such as LEAD ((See, Liu, & Manning, 2017), (Nallapati, Zhai, & Zhou, 2017)), with the ROUGE results of R-1: 30.33, R-2:12.39, R-L:27.16. The quantitative as well as the human evaluation which the authors conducted demonstrates that RASG improves summarization

performance. The results are comparable to the current state-of-the-art and the proposed hierarchical attention benefits both disentanglement of topics and summary generation.

Another interesting application of text summarizers is fake news detection as done by Esmaelizadeh, Peh and Xu (2019). Firstly, they investigated the performance of standard summarization models, namely: an LSTM encode-decoder framework with one attention mechanism, an LSTM encoder-decoder with attention and pointer-generator mechanisms, an LSTM encoder-decoder with attention, pointer generator and coverage mechanisms and a transformer model. They reviewed their models using the CNN/Daily Mail corpus and evaluated them with the help of the ROUGE metric. The results for their best model (the LSTM encoder-decoder with attention, pointer-generator and coverage mechanisms) were as follows: R-1:39.97, R-2:17.05, R-L:36.36. That model was then used as a feature generator in order to create summaries of a fake news detection dataset. A fake news article would be considered one that has a sensationalistic title that does not reflect the content of the article. Due to this, the article contains more information than the headline which means that a fake news classifier performs better on article contents. The model was evaluated with a fake news dataset (containing articles and headlines) and the results of the evaluation point to the summarization model serving as a feature generator actually increases the accuracy of the framework.

Fabbri, Li, She, Li and Radev (2019) investigated multi-document news article summarization which can be quite challenging due to issues of avoiding redundancy and organization in the output. The authors used Multi-news which is a novel dataset consistent of news articles from various different sources and human-written summaries of the articles. This dataset can be considered the first large-scale corpus on news articles for the purpose of MDS[11]. The authors' model was a hierarchical MMR- attention pointer-generator network and a transformer model which replaces the recurrent layers with self-attention in the encoder-decoder framework called Hi-MAP. MMR (Maximal marginal relevance) is an approach used for combining query-relevance with information novelty in the context of summarization. The model was evaluated on the Multi-News dataset and the ROUGE scores were as follows: R-1: 43.47, R-2:14.89, R-

[11] MDS- Multi-document summarization

L:17.41. Hi-MAP preformed competitively with (Gehermann, Deng, & Rush, 2018) and older multi-document models such as LexRank and TextRank.

Singh and Shashi (2019) also dealt with the task of multi-document summarization and proposed a new hybrid deep learning architecture that can be considered a cascade of abstractive and extractive summarization. The model first performs abstractive summarization using pointer-generator network and produces multiple short summaries. After that, the model performs extractive multi-document summarization using LexRenk to produce the final output summary. The framework was evaluated using the DUC 2004 dataset and ROUGE evaluation metric and the results are as follows: R-1: 43.03, R-2: 7.2, R-3: 1.3 and R-L:28.9. The results extend the current state-of-the-art and exceed the results in (Fabbri, Li, She, Li, & Radev, 2019). However, Fabbri et al (2019) tested on a novel dataset (Multi-news) which might be the cause of the discrepancies.

Another study that dealt with multi-document neural abstractive summarization was done by (Chu & Liu, 2019). Unlike previous studies, the authors used an unsupervised method and they take into account only documents (product or business reviews) without any summary pairs provided. Since the datasets that have paired document-summary examples are rather rare and the models which are trained on them have a downside of not being able to be transferred to multiple domains, the authors proposed an end-to-end neural model architecture for unsupervised abstractive summarization. The model, which they named MeanSum consists of an auto-encoder module that learns representations for each review and constrains the generated summaries to be in the language domain. In the auto-encoder the mean of the representation of the input reviews decodes to a summary-review without having any knowledge of review-specific features. MeanSum is also equipped with a summarization module which generates summaries of the input reviews and ensures that they are semantically alike the input documents. Both the auto-encoder and the summarization module contain an LSTM encoder and a decoder in which the two encoder's weights and the two decoder's weights are tied. They are initialized with the same language model which was pre-trained on the reviews from the dataset. The dataset consisted of customer reviews provided in the Yelp Dataset Challenge (reviews that were rated with a 5 star score and businesses needed to have at least 50 reviews written about them). The method produced the following results on ROUGE: R-1:29.35, R-2:3.52, R-L:15.97. The authors also

conducted a human evaluation where the raters had to rate 100 summaries on a scale of 1 (very poor) to 5 (excellent) based on how well the sentiment of the summary aligns with that of the review, how well the information is summarized and how fluent the summary is. When it came to sentiment, the raters evaluated the summary produced by MeanSum with 3.91 points on average and 3.83 points for the informativeness of the summaries (they rated an extractive model higher when it came to informativeness – 3.85). Furthermore, the human evaluators rated the quality of the summaries based on five dimensions previously used in DUC-2005 and MeanSum outperformed extractive methods and random reviews when it came to grammar, referential clarity and structure and coherence.

MacAvaney, Sotudeh, Talati, Cohen, Goharian and Filice (2019) explored another important application of summarization – summarization of medical reports, which can aid decision-making in medicine, save a clinician's time and reduce errors by allowing for quicker analysis of medical cases. The authors aimed at creating more complete summaries which are accurate and concise. They proposed a seq2seq abstractive, domain-specific summarization model which they applied to a dataset of radiology reports. The model was based on a pointer-generator network with an additional BiLSTM used to encode an additional ontology. They employed two ontologies – UMLS and RadLex. This model outperformed the current state-of-the art and human evaluation conducted by a radiologist confirmed that the model is detailed and accurate. The RadLex model achieved the best results: R-1:38.42, R-2:23.29, R-L:37.02. Furthermore, expert human evaluation was conducted in which a domain expert (radiologist) evaluated 100 reports amongst which some manually written, some generated using PG (pointer-generator) and some generated using PG with RadLex. The radiologist scored the report in terms of readability, accuracy and completeness on a scale from 1 to 5.The model with RadLex proved to be nearly as accurate as human-written summaries, making critical errors in only 5% of the cases, while the PG model made errors in 8% of the cases. This is promising, because ATS greatly improves the speed and quality of diagnostics and decision-making in radiology.

Another compelling application of summarization lies in summarizing video material, since nowadays a huge amount of information on the Internet is in shape of videos. Many different platforms for video sharing have become increasingly popular (such as YouTube, Vimeo, DailyMotion, Twitch) and an enormous amount of user-made videos is being posted daily, many

being instructional – How2 Videos. The authors presented a model of multimodal abstractive summarization of How2Videos, attempting to fill the gap of many videos not having text meta-data associated with them or the existing ones are not informative enough. They aim to generate short textual summaries that describe the most relevant points of the video, so that the viewers by simply reading the description can get a clear idea about the topic of the video. They used the How2 dataset which contains human annotated video summaries of various topics (cooking, sports, activities, music etc.). The authors produce models which generate abstractive descriptions for the video content using the user-generated transcriptions and the output of ASR systems. Furthermore, they propose a novel evaluation metric which they named Content F1. The videos were represented by features extracted from a pre-trained action recognition model (ResNeXt-101 3D CNN) and ASpIRE and EESEN were used for distant-microphone conversational speech recognition. The authors used various summarization models - RNN seq2seq mode, pointer-generator model and a hierarchical attention approach. The hierarchical attention approach computes the context vector independently for each of the input modalities, computes the context vector independently for each of the modalities (text and video), after which the context vectors are treated as states od another encoder and a new vector is computed. The authors used ROUGE-L to evaluate their models, as well as Content F1 which is the F1 score of the content words in the summaries based over a monolingual alignment. The multi-modal hierarchical attention model achieved the best results on ROUGE-L: 54.9 and Content F1:48.9. The PG and S2S models preform comparatively to each other (50.2 and 53.9 respectively on R-L for text-only on the complete transcript). Human evaluation was also conducted and the model achieved 3.89 for informativeness, 3.74 for relevance, 3.85 for cohesion and 3.94 for fluency (on the scale 0-5). This study is a great insight into what can be achieved with ATS and ASR and how it can save users' time and elevate their experience.

Another study dealing with abstarctive summarization of videos was presented by Dilawari & Ghani Khan (2019). They produced the ASoVS (Abstractive Summarization of Video Sequences) model that leverages deep neural networks to generate the description of the video and an abstractive summary of the contents of the video. The first task is description generation and the input to the video description model is a video clip divided into sequences of video frames. For each frame, the features are mined using a CNN after which they are clustered to represent information in form of sentences. For acquiring all the visual features, the authors used

VGG-16 which is pre-trained on ImageNet. After all the features are acquired, they are extracted from the CNN frame by frame and average pooling is conducted. The features are then passed to into a bidirectional LSTM. For the generation of the abstractive summary, the architecture consistent of a unidirectional LSTM encoder and a unidirectional LSTM decoder combined with the pointer generator model. The ROUGE results on the CNN/DM dataset were as follows: R-1:40.21, R-2:17.64, R-L:36.89. Furthermore, human evaluators rated the video description model as having 3.89 out of 5 points on the scale of informativeness, 4.05 for conciseness, 3.98 for readability and 86% for correctly identified clips (if the video description matches the video clip). They rated the abstractive summarization model as having 3.55 on the scale of informativeness, 3.52 on the scale of conciseness and 3.92 on the scale of readability. This study proved that by using automatic summarization for creating having short video description as well as video summaries can make the process of selecting a video shorter and can be less biased than human-made summaries.

Zhang, Li, Wang, Fang and Xiao (2019) claimed that a hierarchical CNN framework is more efficient than the conventional RNN seq2seq models when it comes to the task of abstractive summarization. They created a convolutional seq2seq model based on CNN to generate representations of the input text. They stack CNN layers in order to alleviate the issue of CNNs encoding only fixed-size contexts. The multilayer CNNs create hierarchical representations of the input and offer a shortcut to expressing long sequences in parallel. The authors use two CNNs on the source text in order to draw a summary at the word and sentence level respectively and use a hierarchical attention mechanism which they apply on both levels simultaneously. The word level is calculated first, and then re-weighted with respect to the sentence-level attention. To solve the issue of OOV words, the authors apply a copying mechanism to seq2seq, with the task of extracting them. The datasets used for training are Gigaword and the DUC corpus as well as a collection of news stories from the CNN/DM dataset. The results based on the ROUGE metric on the Gigaword dataset are as follows: R-1:37.95, R-2:18.64, R-L:35.11; the results on the DUC corpus are R-1:29.74, R-2:9.85 and R-L:25.81 and the results on the new testing CNN/DM corpus are R-1:42.04, R-2:19.77 and R-1:39.42. The model (CNN-2sent-hieco-RBM) outperforms models such as RASElman (Chopra, Auli, & M., 2016) and GAN (Linqing, Yao, Min, Qiang, Jia, & Hongyan, 2017) which proved that the model is effective and efficient.

The two most popular abstractive models for automatic summarization are the sequence-to-sequence model and the LSTM bidirectional model. In their paper, (Parmar, Chaubey, Bhatt, & Lokare, 2019) compare the performance of the two methods and evaluate them using the BLEU and ROUGE metrics on Amazon reviews and CNN news datasets. The sequence to sequence model produces the best BLEU results on Amazon reviews (26.25), while it scores 10.63 on CNN/DM corpus. The sequence to sequence model produces the best ROUGE scores (R-1: 58.21 and R-2: 20.57).

Gidiotis and Tsoumakis (2019) took on the task of summarizing scientific articles in a structured way, so that the summary itself follows the structure of the original document. They named their summarizer SUSIE (Structured Summarizer) and the model leverages on the XML structure of the articles and abstracts in order to split each article into multiple training examples and train summarization models that learn to summarize each section independently. The usual structure of an academic article is the IMRD (introduction, methods, results and discussion) with additional sections like conclusion and literature. The authors annotated the sections of the article and the abstract by looking for specific keywords in the header of each section (e.g. method, techniques and methodology are annotated as methods). After the annotation process is done, each section is paired with the full text in the corresponding section of the abstract thus creating one training example per section. In this way, a summarization model can be trained on sections, which makes the process easier since the input and the output are shorter sequences. The authors combined SUSIE with three popular summarization models – attention sequence-to-sequence, pointer-generator and pointer-generator+ coverage. They created the PMC-SA dataset (PMC Structured Abstracts) from scientific articles from the field of biomedicine that can be found on PubMed Central (a digital repository). The best results on ROUGE were achieved with the pointer-generator and coverage summarization model combined with SUSIE and the results were R-1:37.1, R-2:14.6 and R-L: 32.6. The authors found that training SUSIE with PMC-SA greatly improves the quality of the summaries and that SUSIE improved the score of the flat summarization approach for all three models by 4 ROUGE points.

Moroshko, Feigenblat, Roitman and Konopicki (2019) proposed a mixed, extractive-abstractive model which mimics the behavior of a human editor, thus it is called an Editorial network. The editorial network iterates over the sentences given by the extractor and has three choices – to

leave the sentence unchanged, to rephrase or to reject the sentence. The extractor consists of an encoder that uses hierarchical representation and a sentence selector which uses a pointer-generator network. The abstractor is an encoder-alignment-decoder with a copy mechanism which is not only applied on a single extracted sentence but on a lexical chunk of three consecutive sentences, thus the abstractor gains on context. The Editorial network approach was evaluated on CNN/DM with the help of the ROUGE metric and achieved the following scores: R-1:41.42, R-2:19.03, R-L:38.36, thus outperforming various abstractive and extractive methods such as (See, Liu, & Manning, 2017), (Guo, Pasunuru, & Bansal, 2018), (Pasunuru & Bansal, 2018), (Gehermann, Deng, & Rush, 2018), (Chen & Bansal, 2018), etc. The authors add that 56% of the Edit Net's decisions were to abstract and 18% to reject, while only 33% of the extracted sentences were kept in the same form. They conclude that the best choice for ATS would be to combine the abstractive and extractive methods.

Nayeem, Fuad and Chali (2019) used a neural seq2seq encoder decoder model to create a novel abstractive sentence compression model which paraphrases the sentences of the input, improving informativeness and abstractiveness of the generated summary. They named the model DPC (Diverse Paraphrastic Compression model). The model is based on Neural Machine Translation, which it used to translate from the source sentence to an abstractive compression with diversity. The encoder is a bidirectional 3- layer stacked GRU and the COPYNET model is assed to the decoder in order to integrate the word generation in the decoder with a copying mechanism which chooses word sequences in the input sequence and places them appropriately in the output sequence. The model implicitly learns to paraphrase and generates paraphrases from the data itself. Using the fastText embedding, the authors create an alignment table for OOV words to the words inside of the vocabulary and calculate the cosine distance between fastText word vectors to achieve word-to-word alignment. To generate more diverse output sentences, the authors use diversity-promoting beam search. For evaluation, they used the GIGAWORD dataset along with several evaluation metrics: BLEU, SARI, METEOR-E and Compression Ratio (CR). SARI (system output against references and against the input sentence) (Xu, Napoles, Chen, Pavlick, & Chris, 2016) is a novel evaluation metric that measures the appropriateness of the words that are added to the system, deleted from it or maintained by it. It calculates the overlap between the input, the human references and the output, by taking into account recall and precision. Furthermore the authors used Copy Rate to establish how many tokens were copied to the

abstract sentence from the source sentence without paraphrasing (if the core is 100 then no paraphrasing was done). The results were as follows: BLEU: 54.9, SARI: 39.3, METEOR-E 0.41, CR:0.47 and the result for Copy Rate was 84.5. The model outperformed several seq2seq, pointer generator and tree-to-tree transduction models. The results prove that this model achieves a higher level of abstractiveness in the produced summary.

Zhang, Xu and Wang (2019) leveraged BERT in constructing an abstractive summarization model. They used it to encode the input sequence into context representations. Furthermore, the decoder in their model functions in two stages – first of which uses a Transformer-based decoder to generate a draft output sequence while in the second stage each word of the output sequence is mask and fed to BERT which then generates a draft representation. Combining the representation and the input sequence, the Transformer-based decoder predicts the word for each masked position. The model was evaluated on the CNN/DM dataset as well as the NYT corpus and the ROUGE metric was used for evaluating the summaries. The model achieved the following results: R-1:41.71, R-2:19.49, R-L: 38.79 on the CNN/DM corpus and R- 1:45.33, R-2:26.53 on the NYT corpus. The results for R-1 and R-2 on the CNN/DM corpus were comparable to results achieved by DCA (Celikyilmaz, Bosselut, He, & Choi, 2018), but outperform the model on R-L. Furthermore, the model outperforms numerous extractive and abstractive models such as (See, Liu, & Manning, 2017), (Narayan, Cohen, & Lapata, 2018), (Chen & Bansal, 2018), (Zhou, Yang, Wei, Huang, Zhou, & Zhao, 2018), (Gehermann, Deng, & Rush, 2018). The authors add that the model can be used in most NLP tasks such as MT, question generation and paraphrasing.

In a recent study, Lebanoff (2019) claimed that there is a need the bridge the gap between sentence selection and sentence fusion. When writing summaries, humans tend to combine multiple sentences from the original text and merge them in a single, concise summary sentence. However, mechanisms that deal with sentence selection when it comes to automatic summarization, work with single sentences rather than combinations of them. Therefore, the author proposes a framework which attempts to mimic the human process by selecting singles or pairs of sentences and fusing them (compressing) to produce a summary sentence. The authors attempted to create a model which can determine if a single sentence or a sentence pair should be selected to produce a summary sentence. They used the BERT architecture to learn instance representations, as BERT can encode singletons and pairs indiscriminately. It constructs an input

sequence which is then fed to a multi-layer and multi-head attention architecture which then builds deep contextual representations. BERT is then used to fine-tune the representations with an additional output layer. For the purpose of automatic summarization, the authors employ the MMR principle to select a set of non-redundant instances (single sentences and sentence pairs). The MMR principle prevents the system to choose instances that are overly similar to the ones already in the summary, thus resolving the issue of redundancy. A pointer-generator network is used to create an abstractive summary and when trained on document-summary pairs, the model removes unnecessary content and merges multiple sentences together. The network is trained with instances stemming from human summaries, while at test time it receives an instance from BERT and generates a summary sentence. The authors used Xsum dataset (Narayan, Cohen, & Lapata, 2018), CNN/DM and the DUC-04 dataset. The authors compared their model with SumBasic, KL-Sum and LexRank using the ROUGE metric. The results on CNN/DM dataset were R-1:41.13, R-2: 18.68, R-L: 37.75; the results on XSUM were R-1:23.53, R-2:6.48, R-L:19.75; the results on DUC-04 were R-1:30.49, R-2:5.12, R-SU:9.05. This relates to the BERT extractive model, while in some occasions the abstractive variant outperforms these results which relates to the amount of sentence pairs selected, as selecting more pairs than singletons tends to affect the abstractor negatively. It outperformed models such as LexRank (Erkan & Radev, 2004), SumBasic (Vanderwernde, 2007) and KLSumm (Haghigi & Vanderwende, 2009) as well as Extract and Rewrite (Song, Zhao, & Liu, 2018). The results are considered promising for further improvement of ATS models.

Khandelwal, Clark, Jurafsky and Kaiser (2019) used a pre-trained decoder-only network equipped with a Transformer LM which encodes the source and generates the abstractive summary. Instead of using ELMo (which trains the language model in both directions) or BERT (trains a bidirectional word imputation model) the authors trained a unidirectional LM. The model consists of a Transformer encoder that reads the input, a Transformer decoder that generates the summary and an encoder-decoder attention mechanism that enables the decoder to attend the encoder states for output generation. The encoder-decoder model is simplified by considering summarization a language modeling task by appending each summary to its source article with a delimiter and training a Transformer on this data. The authors compare three methods of pre-training the LM – encoder only, decoder only or both. The models are pre-trained on WikiLM, a 2-billion-word corpus based on Wikipedia. For evaluation purposes, the authors

use the CNN/DM corpus and evaluate the results using the ROUGE metric. Pre-training improves efficiency when done on both the encoder and the decoder, but the improvements are much greater when using the Transformer ML. The pre-trained Transformer ML model achieved the following results: R-1:39.65, R-2:17.74 and R-L:36.85. The authors also evaluated models by (Celikyilmaz, Bosselut, He, & Choi, 2018) and (Gehermann, Deng, & Rush, 2018) and their own model achieved the highest ROUGE results. The single, pre-trained Transformer LM for seq-to-seq tasks seems to simplify the model architecture and proves to be efficient.

Hoang, Bosselut, Celikyilmaz and Choi (2019) worked on adapting the transformer language models as text summarizers in two ways – using source embeddings and domain-adaptive training. The Transformer had been pre-trained on a large corpus and based on the GPT model (Radford, Wu, Child, Luan, Amodei, & Sutskever, 2019). The authors contributed with the addition of domain-adaptive training and end task training. Domain-adaptive training is used to adapt the transformer summarization model to the language distribution of newswire text and end task learning is training the model to be able to produce a summary from a given a document by maximizing the conditional loglikelihood of producing the correct output tokens from the set of source tokens. The authors tested their model on the CNN/DM, XSum and Newsroom datasets. The results show that the model slightly underperforms on CNN/DM (R-1:37.96. R-2:17.36, R-L: 35.12) in comparison to models such as PGen (See, Liu, & Manning, 2017), RougeSAL (Pasunuru & Bansal, 2018), Bottom-Up Summ (Gehermann, Deng, & Rush, 2018), DCA (Celikyilmaz, Bosselut, He, & Choi, 2018), CopyTransformer (Gehermann, Deng, & Rush, 2018) and rnn-ext+RL (Chen & Bansal, 2018). However, human evaluation showed that the Hoang et al model was preferred over the previously mentioned models, based on non-redundancy, coherence, focus and overall impression. On the XSum corpus the model had the following results: R-1: 36.76, R-2: 14.93 and R-L: 29.66 and on the largest corpus – Newsroom – the results were R-1: 40.87, R-2: 28.59, R-L: 37.62. The authors conclude that possibly because of the length of summaries, ROUGE might not be consistent with human evaluation.

Author	Type of summarization	Summarization method	Dataset used	Evaluation measure	Features	Results
(Nallapati, Zhai, & Zhou, 2017)	Extractive, abstractive summarization	SummaRuNNer	CNN/DailyMail corpus (286722 training documents, 13362 validation documents and 11482 test documents) DUC 2002 single-document summarization dataset	Rouge-1 Rouge-2 Rouge-L	Bidirectional GRU-RNN	Abstractive summarization (joint CNN/Daily Mail Corpus): 37.5, 14.5 and 33.4 (DUC 2002): 46.6 , 23.1 and 43.03 . Extractive summarization: (joint CNN/Daily Mail Corpus): 39.6, 16.2 and 35.3 (DUC 2002): 44.8, 21.0 and 41.2
(Verma & Daniel, 2017)	Generalized model of summarization	DocSumm (TF/IDF)	DUC 2001-2002 datasets – 533 unique documents	Rouge-1 Rouge-2 Rouge-LCS	Single-document summarization, unification of extractive vs abstractive, syntactic vs semantic	ROUGE- 1 44.0 ROUGE-2 27.2 ROUGE-LCS 29.5
(Hua & Wang, 2017)	Abstractive summarization	Neural summarization model pre-trained on extractive	The New York Times Annotated Corpus (100824 articles from the domain of news and	Rouge-2 Rouge-L BLEU	Sequence to sequence model with a pointer generator network, evaluation of domain	News articles: ROUGE-2 24.2 ROUGE-L

		summaries	51214 from the domain of opinion)		effects on summarization	34.5 BLEU 22.4 Opinion articles: ROUGE-2 19.9 ROUGE-L 31.8 BLEU 14.22
(Paulus, Xiong, & Socher, 2017)	Abstractive summarization	Neural network model with intra-attention (training method includes reinforcement learning and supervised word prediction)	CNN/Daily Mail NYT dataset	ROUGE-1 ROUGE-2 ROUGE-L Human evaluation (scale 1-10)	Reinforcement learning, Intra-attention model (for repetitive and incoherent phrases) Inter-temporal attention model Pointer mechanism	CNN/Daily Mail: R-1: 39.87 R-2:15.82 R-L: 36.90 NYT dataset: R-1: 42.94 R-2: 26.09 Human evaluation Readability: 7.04 Relevance: 7.45
(Nema, Khapra, Laha, & Ravindran, 2017)	Query-based abstractive summarization	Neural encode-attend-decode model with soft LSTM diversity-based attention	Debatapedia (663 debates)	ROUGE-1 ROUGE-2 ROUGE-L	Query attention model	R-1: 41.26 R-2: 18.75 R-L:40.43

		model				
(Li, Lam, Bing, & Wang, 2017)	Abstractive summarization	DRGN	Gigaword DUC 2004 LCTS	ROUGE-1 ROUGE-2 ROUGE-L	Latent structure modeling component Variational auto-encoders	Gigawords: R-1: 36.27 R-2: 17.57 R-L:33.62 DUC 2004 R-1: 31.79 R-2: 10.75 R-L:27.48 LCTS R-1: 36.99 R-2: 24.15 R-L:34.21
(See, Liu, & Manning, 2017)	Abstractive summarization	Pointer-Generator neural network	CNN/Daily Mail	ROUGE-1 ROUGE-2 ROUGE-L METEOR	Pointer-generator network, solves issue of OOV words	R-1:39.53 R-2:17.28 R-L:36.38 METEOR: 17.32 on Meteor in exact match mode 22.21 in full mode
(Mehta, Aurora, & Majumder, 2018)	Extractive summarization	LSTM sentence encoder, topic modeling based context encoder, attention module and binary	27801 scientific articles from the ACL anthology	ROUGE-1 ROUGE-2 ROUGE-3 ROUGE-4	Context embedding technique	R-1: 34.4 R-2: 9.0 R-3: 4.2 R-4: 2.7

		classifier				
(Narayan, Papasarantopoulos, Lapata, & Cohen, 2017)	Extractive summarization	SideNet (neural summarizer taking account of side information)	Single document summarization – CNN dataset	ROUGE-1 ROUGE-2 ROUGE-3 ROUGE-4 ROUGE-L, Human evaluation	CNN sentence encoder, RNN (LSTM) document encoder and sentence extractor	ROUGE: R-1: 54.2 R-2: 21.6 R-3: 12.0 R-4: 7.6 R-L: 48.1 Human evaluation: Human annotated highlights – 1st place, SideNet- 2nd place
(Tarnpradab, Liu, & Hua, 2017)	Extractive summarization of forum threads	supervised thread summarization approach	Threads taken from TripAdvisor and UbuntuForums	ROUGE-1 ROUGE-2 ROUGE-L Sentence-level precision, recall, f-scores	Neural hierarchical attention networks	TripAdvisor dataset : R-1: 37.8 R-2:14.4 R- 32.5, P – 34.4 and F- 33.4 UbuntuForums: R-1:37.6 R-2: 14.4 R-33.9 P-33.8 F- 33.8
(Sinha, Yadav, & Gahlot, 2018)	Extractive single-document summarization	feedforward neural networks	DUC 2002 dataset	ROUGE-1 ROUGE-2	Simple implementation, less memory complexity than seq2seq models	R-1:55.1 R-2: 22.6
(Zhou, Yang, Wei, Huang, Zhou, & Zhao, 2018)	Extractive summarization	NEUSUM	CNN/Daily Mail	ROUGE-1 ROUGE-2 ROUGE-L	Sentence scoring and selection in one step BiGRU encoder,	R-1: 41.59, R-2:19.01 R-L: 37.98

					GRU decoder, MLP sentence scoring	
(Zhang, Tan, & Wan, 2018)	Extractive summarization	Latent variable extractive model	CNN/Daily Mail đ	ROUGE-1 ROUGE-2 ROUGE-L	Latent variable model improves the extractive model	R-1: 41.05 R-2:18.77 R-L:37.54
(Narayan, Cohen, & Lapata, 2018)	Extractive summarization	REFRESH (Reinforcement Learning-based Extractive Summarization)	CNN/Daily Mail	ROUGE-1 ROUGE-2 ROUGE-L	Hierarchical document encoder (RNN-LSTM), reinforcement learning	CNN dataset R-1:30.4 R-2:11.7 R-L:26.9 Daily Mail R-1:41.0 R-2:8.8 R-L: 37.7
(Wu & Hu, 2018)	Extractive summarization	RNES (Reinforced Neural Extractive Summarization)	CNN/ Daily Mail	ROUGE-1 ROUGE-2 ROUGE-L	Reinforcement learning, ROUGE reward, coherence model	R-1: 40.95 R-2: 18.63 R-L: 37.41
(Al-Sahabi, Zuping, & Nadher, 2018)	Extractive summarization	HSSAS-Hierarchical Structured Self-Attentive Model for Extractive Document Summarization	CNN/Daily Mail DUC-2002	ROUGE-1 ROUGE-2 ROUGE-L	hierarchical structured self-attention mechanism for creating embeddings	DUC-2002 R-1:52.1 R-2:24.5 R-L: 48.8 CNN/DM R-1:42.3 R-2: 17.8

						R-L: 37.6
(Liao, Lebanoff, & Liu, 2018)	Abstractive summarization (multidocument)	Abstract Meaning Representation (AMR)	DUC-2004 TAC - 2011	ROUGE-1 ROUGE-2 ROUGE-3 ROUGE-SU4	AMR, multidocument	DUC 2004 R-1: 37.8 R-2: 6.6 R-SU4:11.8 TAC 2011: R-1: 41.1 R-2: 8.5 R-SU4:13.5
(Dohare, Gupta, & Karnick, 2018)	Abstractive summarization	Abstract Meaning Representation (AMR)	CNN/Daily Mail	ROUGE-1 ROUGE-2 ROUGE-L	Meta nodes, Co-reference resolution	R-1:40.9 R-2:16.7, R-L: 29.5
(Hardy & Vlachos, 2018)	Abstractive summarization	NLG model	Proxy Report section from the AMR dataset	ROUGE-1 ROUGE-2 ROUGE-L	Seq2seq model for NLG	R-1: 42.3 R-2: 20.3 R-L: 31.4 -
(Kodaira & Komachi, 2018)	Abstractive summarization (summaries consistent of 3 sentences)	Model by (Nallapati, Zhai, & Zhou, 2017)	214120 pairs of articles and summaries by Japanese LiveNews	ROUGE-1 ROUGE-2 ROUGE-L	Dataset in which articles consist of only three bullet points	R-1: 49.48 R-2 : 29.15 R-L 35.82
(Xie, Li, Ren, & Zhai, 2018)	Extractive summarization (abstractive and	Seq2seq dual attentional model	CNN/Daily Mail	ROUGE-1 ROUGE-2 ROUGE-L	WordNet based sentence ranking, leading three method	Leading three: R-1: 39.41 R-2:17.30

	extractive methods at sentence level)				biLSTM encoder and dual ettention decoder (uniLSTM)	R-L: 35.92 WordNet: R-1: 39.32 R-2:17.15 R-L: 36.02
(Arumae & Liu, 2018)	Extractive summarization	Bidirectional LSTM encoder, attention mechanism	CNN	ROUGE-1 ROUGE-2 ROUGE-L	Novel question-focused rewards, reinforcement learning	R-1:31.7 R-2:11.6, R-L: 21.5
(Gehrmann, Layne, & Dernoncourt, 2019)	Extractive summarization	Encoder-decoder model using unsupervised word representations (BERT, ELMo)	CNN/ Daily Mail Google sentence compression dataset	ROUGE-1 ROUGE-2 ROUGE-L Human evaluation	Deletion-based compression based on Semi-Markov Conditional Random Field	R-1: 30.2 R-2: 12.2 R-L: 26.45 Human evaluation – average of 68.25% of questions answered correctly
(Xu & Durrett, 2019)	Extractive summarization	JECS (Joint Extractive and Compressive Summarizer)	CNN/Daily Mail NYT	ROUGE-1 ROUGE-2 ROUGE-L	Compression rules and parse tree to compress chunks, seq2seq model, ELMo	CNN/DM: R-1: 40.3 R-2:17.6 R-L:36.4 NYT R-1: 44.3 R-2: 25.5 R-L: 37.1
(Liu, Cheung, & Louis, 2019)	Extractive summarization	NEXTSUM	NYT dataset	ROUGE-2	The model captures the internal structure	CRIME R-2: 28.1

					of the summary, sentence prediction +sentence generation, divides articles by domains	ASSASSINATION R2: 24.1 BOMBS R2: 25.0
(Liu Y. , 2019)	Extractive summarization	BERTSUM	CNN/DM NYT dataset	ROUGE-1 ROUGE-2 ROUGE-L	BERT leveraged for ATS Model combined with a Transformer	CNN/DM R-1: 43.25 R-2: 20.24 R-L:39.63 NYT R-1: 46.66 R-2:26.35 R-L: 42.62
(Ga & Hu, 2019)	Extractive summarization	BERT summarization	CNN/DM	ROUGE-1 ROUGE-2 ROUGE-L Human evaluation	BERT text encoder	R-1:37.30 R-2:17.05 R-L: 34.76 Relevance and readability – 85/100
(Khan, Qian, & Naeem, 2019)	Extractive summarization	K-mean + TF/IDF summarization	Dataset consisting of headlines, summaries and articles taken from Hindu times, Indian times and The Guardian	BLEU	K-means clustering, TFIDF model	Elbow method Doc1 – 0.39 Silhouette method Doc1- 0.42
(Liu, Titov, & Lapata, Single	Extractive summarization	SUMO (Structures	CNN/Daily Mail NYT dataset	ROUGE-1 ROUGE-2	Dependency discourse tree,	DM R-1:42.0

Document Summarization as Tree Induction, 2019)	(single document)	Summarization Model)		ROUGE-L	Transformer architecture	R-2: 19.1 R-L: 38.0 NYT R-1: 42.3 R-2:22.7 R-L: 38.6
(Azunre, et al., 2018)	Abstractive summarization	DUKE (Dataset Understanding via Knowledge-based Embeddings)	CKAN tabular dataset OpenML tabular dataset Data.world dataset	Manual grading	Abstractive summarization of tabular data, including text in headers columns or supporting metadata	CKAN tabular dataset: accuracy (high for first two, medium for second two) OpenML tabular dataset(high for first two, medium for second two) Data.world dataset (medium for first two, high for second two)
(Celikyilmaz, Bosselut, He, & Choi, 2018)	Abstractive summarization	DCA MLE+SEM+RL	CNN/Daily Mail NYT dataset	ROUGE-1 ROUGE-2 ROUGE-L	Multi-agent neural encoder (bi-directional LSTM) and neural decoder (single-layer LSTM), multi-agent pointer network	CNN/Daily Mail: R-1:41.69 R-2: 19.47 R-L: 37.92 NYT dataset: R-1:48.08 R-2: 31.19 R-L: 42.33
(Gehermann, Deng, & Rush, 2018)	Abstractive summarization	Bottom-up	CNN/Daily Mail NYT dataset	ROUGE-1 ROUGE-2 ROUGE-L	Neural network approach with content selector as a bottom-up attention step	CNN/Daily Mail: R-1:41.22 R-2: 18.68 R-L: 38.34

						NYT dataset: R-1:47.38 R-2: 31.23 R-L: 41.81
(Al-Sabahi, Zuping, & Kang, 2018)	Abstractive summarization	Bidir_Rev_Cov	CNN/Daily Mail NYT dataset	ROUGE-1 ROUGE-2 ROUGE-L	Bidirectional RNN model (encoder and decoder as LSTM), bidirectional beam search	CNN/Daily Mail: R-1: 42.6 R-2:18.8 R-L: 38.5
(Song, Zhao, & Liu, 2018)	Abstractive summarization	Struct+2way+word	Gigaword	ROUGE-1 ROUGE-2 ROUGE-L Likert scale	bi-directional LSTM as an encoder and LSTM decoder with attention mechanism, copy mechanism and 2-way combination (+word/relation)	ROUGE R-1: 35.47 R-2:17.66 R-L: 33.52 LIKERT 3.0 on informativeness 3.4 on fluency 3.1 on faithfulness.
(Pasunuru & Bansal, 2018)	Abstractive summarization	RougeSal+Ent	CNN/Daily Mail DUC-2002 Multi-NLI SNLI	ROUGE-1 ROUGE-2 ROUGE-L CNN/Daily Mail Cloze Q&A setup	sequence-to-sequence single-layer bidirectional encoder and an unidirectional LSTM decoder with attention, pointer-copy and coverage mechanisms (ROUGESal and	R-1:40.43 R-2:18.00 R-L:37.10 CNN/Daily Mail Cloze Q&A setup 60.66%, 59.36%,60.67% and 64.66% Abstractiveness: 2-gram:2.63, 3-gram:6.56, 4-gram:10.26

					Entail rewards)	
(Guo, Pasunuru, & Bansal, 2018)	Abstractive summarization	Soft-layer multitask with entailment and question generation	CNN/Daily Mail Gigaword Multi-NLI SNLI	ROUGE-1 ROUGE-2 ROUGE-L	sequence-attention-sequence model with a 2-layer bidirectional LSTM encoder and a 2-layer unidirectional LSTM decoder + question generation and entailment generation	CNN/Daily Mail corpus R-1:39.81 R-2:17.64 R-L: 36.54 METEOR: 18.54 Gigaword corpus: R-1:35.98, R-2:17.76, R-L:33.63
(Cohan, et al., 2018)	Abstractive summarization of long documents	Hierarchical LSTM encoder and discourse-aware LSTM decoder	arXiv.org PubMed.com	ROUGE-1 ROUGE-2 ROUGE-3 ROUGE-L		arXiv dataset: R-1: 35.80 R-2: 11.05 R-3: 3.62 R-L: 31.80 PubMed dataset R-1: 38.93 R-2: 15.37 R-3: 9.97 R-L: 35.21
(Kryściński, Paulus, Xiong, & Socher, 2018)	Abstractive summarization	Discourse-aware attention model for abstractive summarization	CNN/Daily Mail Percentage of novel n-grams	ROUGE-1 ROUGE-2 ROUGE-L	Maximum likelihood + Reinforcement learning with novel ROUGE reward and external Language model	R-1: 40.72 R-2: 15.95 R-L: 38.14 Novel n-gram percentage: NN-1.: 3.19 NN-2: 22.79 NN-3: 39.9

						NN-4:50.61
(Zhang, Tan, & Wan, 2018)	Abstractive summarization	Seq2seq for multi-document summarization	DUC-2004	ROUGE-1 ROUGE-2 ROUGE-SU	Improved SinABS by adding a document set encoder	R-1: 36.7 R-2: 7.83 R-SU: 12.4
(Baumel, Eyal, & Elhadad, 2018)	Abstractive query-based summarization	RSA QFS (Relevance Sensitive Abstractive QFS)	DUC- 2005,2006,2007	ROUGE-1 ROUGE-2 ROUGE-L	Pointer-generator network with query relevance	DUC 2005 R-1: 39.82 R-2:6.98 R-L: 15.73 DUC 2006 R-1: 42.89 R-2:8.73 R-L: 17.75 DUC 2007 R-1: 43.92 R-2:10.13 R-L: 18.54
(Li, Bing, & Lam, 2018)	Abstractive summarization	AC-ABS (Actor-critic abstractive summarization)	GIGAWORD DUC-2004 LCSTS	ROUGE-1 ROUGE-2 ROUGE-L	Reinforcement learning actor-critic approaches	Gigaword R-1: 36.05 R-2: 17.35 R-L: 33.49 DUC-2004 R-1: 32.03 R-2: 10.99 R-L: 27.86 LCSTS R-1: 37.51

						R-2: 24.68 R-L: 35.02
(Amplayo, Lim, & Hwang, 2018)	Abstractive summarization	S2s+att+E2T (CNN+SD)	GIGAWORD CNN	ROUGE-1 ROUGE-2 ROUGE-L	Entity2Topic module attachable to seq2seq model (transforms list of entities into a vector representation of the topic)	Gigaword R-1: 37.04 R-2:16.66 R-L: 34.93 CNN R-1: 31.9 R-2: 10.1 R-L: 23.9
(Chen & Bansal, 2018)	Abstractive summarization	rnn-ext +abs +RL + rerank	CNN/Daily Mail	ROUGE-1 ROUGE-2 ROUGE-L METEOR	Hybrid extractive-abstractive model with policy-based RL, actor-critic policy gradient Parallel decoding	CNN/Daily Mail: R-1: 40.88 R-2:17.80 R-L: 38.54 METEOR with rerank strategy: 20.38 without rerank: 21.00
(Keneshloo, Ramakrishnan, & Reddy, 2018)	Abstractive summarization	Transfer RL	DUC 2003 DUC 2004 CNN/DM Newsroom	ROUGE-1 ROUGE-2 ROUGE-L	RL framework with self-critic policy gradient approach, transfer learning models	Average score : R-1: 36.21 R-2: 22.25 R-L: 32.81
(Ouyang, Song, & McKeown, 2019)	Abstractive summarization	ABS-so ABS-sw ABS-tl ABS-mix	NYT corpus DUC 2004	ROUGE-1 ROUGE-2 ROUGE-L	Cross-lingual summarization (Swahili, Somali, Tagalog, Arabic)	Somali NYT: R-1: 38.07 R-2:15.76

					Pointer-generator network	R-L: 26.82 Swahili NYT: R-1: 39.96 R-2:17.56 R-L: 30.24 Tagalog NYT: R-1:40.96 R-2:18.91 R-L: 31.14 Arabic DUC 2004 R-1: 29.43 R-2: 7 .02 R-L: 19.98
(Kim, Kim, & Kim, 2019)	Abstractive summarization	MMN (Multi-level memory networks)	TIFU dataset (Reddit discussions)	ROUGE-1 ROUGE-2 ROUGE-L	Novel memory network model MMN, summarization on forum discussions	R-1: 19.0 R-2: 3.7 R-L: 15.1
(Karn, Chen, Chen, Waltinger, & Schütze, 2019)	Abstractive summarization	Hier2hier_fLSTM	Synthetic dataset from a corpus of conventional texts adjusted from the PubMed corpus	ROUGE-1 ROUGE-2 ROUGE-L	Summarization of multi-participant, threaded posting Hierarchical encoder-decoder network with	R-1:41.76 R-2:16.89 R-L: 30.70

					hierarchical attention mechanism	
(Gao, Chen, Li, Bing, Zhao, & Yun, 2019)	Abstractive summarization	RASG (reader aware summary generator)	document-summary-comment pair data from Weibo	ROUGE-1 ROUGE-2 ROUGE-L	Abstractive summaries of news document including user comments Seq2seq model with a reader attention module	R-1: 30.33 R-2:12.39 R-L:27.16
(Esmaelizadeh, Peh, & Xu, 2019)	Abstractive summarization	LSTM encoder decoder with attention mechanism, PG + coverage mechanism	Fake news dataset CNN/DM	ROUGE-1 ROUGE-2 ROUGE-L	Summarization used for fake news detection	R-1: 39.97 R-2:17.05 R-L:36.36.
(Fabbri, Li, She, Li, & Radev, 2019)	Abstractive summarization	Hi-MAP (encoder-decoder based on MM-attention PG network)	Multi-news	ROUGE-1 ROUGE-2 ROUGE-L	Multi-document news article summarization	R-1: 43.47 R-2:14.89 R-L: 17.41
(Singh & Shashi, 2019)	Abstractive summarization	Hybrid deep learning architecture – a cascade of abstractive and extractive	DUC-2004	ROUGE-1 ROUGE-2 ROUGE-L	Abstractive summarization using PG network and then extractive multi-document summarization using	R-1: 43.03 R-2: 7.2 R-3: 1.3 R-L:28.9

		summarization			LexRank	
(Chu & Liu, 2019)	Abstractive summarization	MeanSum	Yelp Dataset Challenge (reviews)	ROUGE-1 ROUGE-2 ROUGE-L Human evaluation	Unsupervised method that takes into account only full document without the summaries Auto-encoder model learns representation for each review	R-1: 29.35 R-2:3.52 R-L: 15.97 Human evaluation: 3.91/5 (informativeness) 3.89 (fluency)
(MacAvaney, Sotudeh, Talati, Cohen, Goharian, & Filice, 2019)	Abstractive summarization	RadLex+PG	Dataset of medical reports	ROUGE-1 ROUGE-2 ROUGE-L Human evaluation (by a domain expert)	RadLex vs UMLS ontology domain-specific model, PG network	R-1: 38.42 R-2:23.29 R-L: 37.02 Human evaluation: Error in 5% OF CASES
(Palaskar, Libovicky, Gella, & Metze, 2019)	Abstractive summarization	Modal abstractive summarizer for HOW2 videos	HOW2 dataset	ROUGE-L Content F1 Human evaluation	Summarization of HOW2Videos (output of ASR system + transcriptions) Novel evaluation metric	ROUGE-L : 54.9 Content F1: 48.9 Human evaluation 3.89/5 informativeness 3.74 relevance 3.85 cohesion 3.94 fluency
(Dilawari & Ghani Khan, 2019)	Abstractive summarization	ASoVS (Abstractive Summarization of Video Sequences)	CNN/DM dataset	ROUGE-1 ROUGE-2 ROUGE-L Human evaluation	Deep neural networks used for generating descriptions of videos and abstractive summary of video	CNN/DM R-1: 40.21 R-2: 17.64 R-L: 36.89 Informativeness 3.98/5

				(informativeness, conciseness, readability, correctness)	contents Encoder-decoder model +PG network	Conciseness 4.05/5 readability 3.98 Correctly identified clip 89%
(Zhang, Li, Wang, Fang, & Xiao, 2019)	Abstractive summarization	CNN-2sent-hieco-RBM (copying mechanism, hierarchical attention mechanism and RBM pre-processing)	GIGAWORD CNN/DM	ROUGE-1 ROUGE-2 ROUGE-L	hierarchical CNN framework +copy mechanism	GIGAWORD R-1: 37.95 R-2 18.64 R-L: 35.11 CNN/DM R-1: 42.04 R-2: 19.77 R-L: 39.42
(Parmar, Chaubey, Bhatt, & Lokare, 2019)	Abstractive summarization	Sequence-to-sequence LSTM bidirectional model	CNN dataset Amazon rewievs	BLEU ROUGE-1 ROUGE-2	Comparison of LSTM bidirectional model and seq2seq model	Seq2seq R-1: 58.21 R-2: 20.57 Bi-LSTM BLEU: 26.25
(Gidiotis & Tsoumakis, 2019)	Abstractive summarization	SUSIE (Structured Summarizer)	PMC-SA dataset (PMS Structured Abstracts)	ROUGE-1 ROUGE-2 ROUGE-L	Model maintains the IMRD structure of academic articles in the produced summary Novel PMC-SA	R-1: 37.1 R-2: 14.6 R-L: 32.6

					dataset Pointer-generator, coverage mechanism	
(Moroshko, Feigenblat, Roitman, & Konopicki, 2019)	Abstractive summarization	Editorial network	CNN/Daily Mail	ROUGE-1 ROUGE-2 ROUGE-L	Extractive-abstractive model that mimics the behavior of a human editor	R-1: 41.42 R-2: 19.03 R-L:38.36
(Nayeem, Fuad, & Chali, 2019)	Abstractive summarization	DPC (Diverse Paraphrastic Compression model)	GIGAWORD	BLEU SARI METEOR-E Compression Ratio Copy Rate	Bidirectional 3-layer stacked GRU encoder, decoder with COPYNET model Model learns to paraphrase fastText embeddings	BLEU: 54.9 SARI: 39.3 METEOR-E: 0.41 CR: 0.47 Copy Rate 84.5
(Zhang, Xu, & Wang, 2019)	Abstractive summarization	Two-stage + RL	CNN/DM dataset NYT dataset	ROUGE-1 ROUGE-2 ROUGE-L	Using BERT to construct a summarizer Transformer-based decoder functions in two stages	CNN/DM R-1:41.71 R-2:19.49 R-L: 38.79 NYT R-1:45.33 R-2:26.53
(Lebanoff, 2019)	Abstractive summarization	GT-Sing-PairMix (ground truth singletons	Xsum dataset CNN/DM	ROUGE-1 ROUGE-2 ROUGE-L	the model selects singles or pairs of sentences and	CNN/DM R-1:41.13 R-2: 18.68

		and pairs)	DUC-2004		compresses them into one sentence BERT encodes singletons and pairs indiscriminately and fine-tunes representations	R-L: 37.75 XSUM R-1:23.53 R-2: 6.48 R-L: 19.75 DUC-2004 R-1:30.49 R-2: 5.12 R-SU: 9.05
(Khandelwal, Clark, Jurafsky, & Kaiser, 2019)	Abstractive summarization	Transformer LM model	Wiki LM CNN/DM corpus	ROUGE-1 ROUGE-2 ROUGE-L	Transformer encoder-decoder	R-1: 39.65 R-2: 17.74 R-L: 36.85
(Hoang, Bosselut, Celikyilmaz, & Choi, 2019)	Abstractive summarization	Transformer SM	CNN/DM XSum Newsroom	ROUGE-1 ROUGE-2 ROUGE-L	Transformer model with domain-adaptive training	CNN/DM R-1:37.96 R-2: 17.36 R-L: 35.12

6. Analysis

The scientific work reviewed in this paper dates from 2017-2019 (July). The research done dealt with 64 extractive and abstractive summarization approaches, from which a total of 19 papers dealt with the extractive summarization while 44 papers dealt with the task of abstractive summarization and one presented a generalized model of summarization which aimed to produce output that was both extractive and abstractive. These numbers point to the popularity of abstractive summarization in recent times, as many researchers strive to create automatic summaries which would be indistinguishable from human-made summaries. Automatic summaries with a high level of abstractiveness, which are at the same time fluent, concise, informative and readable, have already been produced and novel methods continue to arise to improve on the quality of the automatically generated summaries.

The datasets which were most predominantly used for testing the models are CNN/Daily Mail, the New York Times dataset and the DUC 2001-2004 corpora.

The CNN/Daily Mail dataset contains online news articles which are usually no longer than 781 tokens. The articles are paired with multi-sentence summaries which are on average no longer than 4 sentences or 56 tokens. The dataset consists of 287226 training pairs, 13368 validation pairs and 11490 test article/summary pairs. The corpus was most widely used, in almost 55% of articles reviewed in this thesis.

The New York Times annotated corpus contains over 1.8 million articles and 650000 article summaries written by library scientists, who also tagged around 1500000 of the articles. The articles all stem from the same source – The New York Times magazine and they date from 1987 to 2017.

Frequently used are also the DUC (Document Understanding Conference) datasets which are relatively small and consist of newswire articles paired with human summaries. The benefit of these datasets is that multiple reference summaries are available for each article, which tends to improve the ROUGE score.

The Gigaword corpus is also used and it contains around 10 million documents from seven newswire sources such as The Associated Press and New York Times Newswire Service. The corpus is the largest from the previously mentioned and the most diverse when it comes to domains, but it does not contain summaries, so Gigaword's headlines are usually used for training.

The Newsroom dataset was collected using social media and search engine metadata and it contains 1321995 article-summary pairs with each article having the approximate length of 658 words, while each summary contains 27 words on average.

Other than those commonly used datasets, some interesting work was done using datasets in form of forum threads from TripAdvisor and UbuntuForums, scientific articles collected from arXiv and PubMed as well as the TIFU dataset which contains Reddit discussions and the Amazon dataset containing product reviews. A significant mention is also the HOW2 dataset which contains summaries of How2 videos (output from ASR systems and transcriptions)

The vast majority of the studies were conducted on English datasets (92%), while three articles included Chinese datasets (LCTS and Weibo dataset) and one article dealt with low-resource languages, such as Swahili, Somali, Tagalog and Arabic. All the frequently used datasets such as CNN/Daily Mail, Gigaword, New York Times dataset and others previously mentioned are predominantly in English. There is a significant need for further research based on datasets in other languages, such as French, German, Spanish, etc. as well as low-resource languages such as Croatian and we might expect these advances in the future.

When it comes to summarization methods, the most widely used are the RNNs (Recurrent Neural Networks) – 48% of the articles (41 articles) reviewed in this thesis were based on a seq-2-seq RNN architecture. In 12% of the recent work that was based on the RNN architecture, GRU cells were used, and in 53% the choice were LSTM cells. Convolutional neural networks are not often used for the task of automatic summarization and have been used in two of the previously mentioned articles. However, recently hybrid (pointer-generator) networks have been more recently used (9% of the articles), to overcome the issues of out-of-vocabulary words, speed of training and difficulty of copying words from the source text, to resolve the problem of frequent repetitions. Another improvement is the Transformer model which was used in 4 recent

articles and show a significant improvement in the abstractiveness of the generated summaries as well as provide state-of-the-art results. ElMo was used in two most recent articles, BERT in four, and FastText in one. This opened the gates for further future research that optimized the results of the generated summaries.

7. Conclusion

With the ever-growing amount of information found on the internet the need for automatically made summaries keeps growing exponentially. Automatically made summaries can benefit various fields and not only the everyday users of the internet, but also improve businesses, medical work, forensic investigations and many other fields. Recently there has been much advancement in the quality off the generated summaries, especially with the use of neural networks for the task. Neural networks have made the generation of novel language possible, which significantly improved the abstractiveness and the general quality of the summaries. The generated summaries have in various instances proven to be more readable, concise and informative than human-made summaries, while at the same time saving tremendous amounts of time and resources. Nowadays, the most used model for preforming automatic summarization is the sequence-to-sequence model and Recurrent Neural Networks, although in the recent time architectures that are built around word embeddings, hybrid networks and Transformers achieve the best results. Furthermore, most of the work done with automatic text summarization uses datasets such as CNN/Daily Mail, Gigaword and New York Times, which are all based on newspaper articles. Some interesting achievements have been made by using datasets consistent on forum discussions, reviews and video transcriptions as well as on scientific articles which usually have a set structure. However, there is still a lot more research needed on cross-lingual automatic summarization as well as on creating datasets that are in low-resource languages. The task of automatic summarization is non-trivial, but the results are extremely useful, time-saving and a great aid for decision-making.

Bibliography

(n.d.).

Afantenos, S., Karkaletsis, V., & Stramatopoulous, P. (2005). Summarization from medical documents: a survey. *Artificial Intelligence in Medicine, 33*(2), 157-177.

Allahyari, M. e. (2017). Text Summarization Techniques: A Brief Survey. *International Journal of Advanced Computer Science and Applications(IJACSA), 8*(10), 397-405.

Allen Institute for Artificial Intelligence. (2018). *ELMo*. Retrieved July 2019, from AllenNLP: https://allennlp.org/elmo

Al-Sabahi, K., Zuping, Z., & Kang, Y. (2018). Bidirectional Attentional Encoder-Decoder Model and Bidirectional Beam Search for Abstractive Summarization. *CoRR*.

Al-Sahabi, Zuping, Z., & Nadher, M. (2018). A Hierarchical Structured Self-Attentive Model for Extractive Document Summarization. *IEEE Access*, arXiv:1805.07799v1 .

Amplayo, R. K., Lim, S., & Hwang, S.-w. (2018). Entity Commonsense Representation for Neural Abstractive Summarization. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (p. arXiv:1806.05504). Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics.

Arumae, K., & Liu, F. (2018). Reinforced Extractive Summarization with Question-Focused Rewards. *arXiv*, arXiv:1805.10392v2.

Azunre, P., Corcoran, C., Sullivan, D., Honke, G., Ruppel, R., Verma, S., et al. (2018). Abstractive Tabular Dataset Summarization via Knowledge BaseSemantic Embeddings. *Thirty-fifth International Conference on Machine Learning.* ICML 2018.

Banko, M., & Vanderwende, L. (2004). Using N-Grams to understand the nature of summaries. *HLT-NAACL-Short '04 Proceedings of HLT-NAACL 2004: Short Papers* (pp. 1-4). Boston: Association for Computational Linguistics.

Baralis, E., Cagliero, L., Mahoto, N., & Fiori, A. (2013). GraphSum: Discovering correlations among multiple terms for graph-based summarization. *Information Sciences*, 96-109.

Baumel, T., Eyal, M., & Elhadad, M. (2018). Query Focused Abstractive Summarization:Incorporating Query Relevance, Multi-Document Coverage, and Summary Length Constraints into seq2seq Models. *arXiv*, 1801.07704v2.

Blei, D. e. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research, 3*, 933-1022.

Blunsom, P. e. (2017). *https://github.com/oxford-cs-deepnlp-2017/lectures/blob/master/README.md*. Retrieved July 1, 2019, from GitHub: https://github.com/oxford-cs-deepnlp-2017/lectures/blob/master/README.md

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *arXiv*, arXiv:1607.04606v2.

Brownlee, J. (2017, December 18). *A Gentle Introduction to Exploding Gradients in Neural Networks*. Retrieved June 25, 2019, from Machine Learning Mastery: https://machinelearningmastery.com/exploding-gradients-in-neural-networks/

Carenini, G., Murray, G., & Ng, R. (2011). *Methods for Mining and Summarizing Text Conversations.* Morgan & Claypool Publishers.

Carnegie Mellon University. (2010). *About*. Retrieved August 2019, from Meteor - Automatic Machine Translation Evaluation System: https://www.cs.cmu.edu/~alavie/METEOR/

Celikyilmaz, A., Bosselut, A., He, X., & Choi, Y. (2018). Deep Communicating Agents for Abstractive Summarization. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (pp. 1662-1675). New Orleans: Association for Computational Linguistics.

Chen, Y.-C., & Bansal, M. (2018). Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting. *arXiv*, arXiv:1805.11080v1.

Chopra, S., Auli, M., & M., R. A. (2016). Abstractive Sentence Summarization with Attentive Recurrent Neural Networks. *Conference: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. DOI: 10.18653/v1/N16-1012). ACL.

Chu, E., & Liu, P. (2019). MeanSum : A Neural Model for Unsupervised Multi-Document Abstractive Summarization. *The Thirty-sixth International Conference on Machine Learning.* Lon Beach: ICML 2019.

Cohan, A., Dernoncourt, F., Kim, D. S., Bui, T., Kim, S., Chang, W., et al. (2018). A Discourse-Aware Attention Model for Long Documents. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)* (pp. 615-621). New Orleans: Association for Computational Linguistics.

Collobert, R. e. (2011). Natural Language processing (Almost) from Scratch. *Journal of Machine Learning Research, 12*, pp. 2493-2537.

Collobert, R., & Weston, J. (2008). A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. *Proceedings of the 25th international conference on Machine learning* (pp. 160-167). ACM.

Conroy, J., & O'leary, D. (2001). Text Summarization via hidden Markov Models. *SIGIR '01 Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, (pp. 406-47). New York.

Debatapedia. (2011, November 11). *Debatapedia*. Retrieved August 3rd, 2019, from Welcome to Debatepedia!: http://www.debatepedia.org/en/index.php/Welcome_to_Debatepedia%21

Devlin, J. C. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv*.

Dilawari, A., & Ghani Khan, M. (2019). ASoVS: Abstractive Summarization of Video Sequences. *IEEE Access*.

Dohare, S., Gupta, V., & Karnick, H. (2018). Unsupervised Semantic Abstractive Summarization. *Conference: Proceedings of ACL 2018, Student Research Workshop* (pp. 74-83). Association for Computational Linguistics.

Dong, Y. (2018). A Survey on Neural-Network-Based Summarization Methods. *CoRR*.

Du, B., Wang, Z., Zhang, L., Zhang, L., & Lieu, W. (2015). Exploring representativeness and informativeness for active learning. *IEEE Transactions on Cybernetics, 47*(1), 14-26.

Edmunson, H. (1969). New Methods in Automatic Extracting. *Journal of the ACM, 16*(2), 264-285.

Elman, J. L. (1990). Finding structure in time. *Cognitive science, 14*, pp. 179-211.

Erkan, G., & Radev, D. (2004). LexRank: Graph-based lexical centraility as salience in text summarizarion. *Journal of Artificial Intelligence Research*.

Esmaelizadeh, S., Peh, G. X., & Xu, A. (2019). Neural Abstractive Text Summarization and Fake News Detection. *ArXiv*, arXiv:1904.00788v1.

Fabbri, A. R., Li, I., She, T., Li, S., & Radev, D. R. (2019). Multi-News: a Large-Scale Multi-Document Summarization Dataset and Abstractive Hierarchical Model. *ArXiv*, arXiv:1906.01749v3.

Ga, Y., & Hu, Y. (2019, March). Extractive Summarization with Very Deep Pretrained Language Model. *International Journal of Artificial Intelligence and Applications (IJAIA), 10*(2).

Ganesan, K. (2018). ROUGE 2.0: Updated and Improved Measures for. *CoRR*.

Gao, S., Chen, X., Li, P. R., Bing, L., Zhao, D., & Yun, R. (2019). Abstractive Text Summarization by Incorporating Reader Comments. *arXiv*, arXiv:1812.05407v1.

Gehermann, S., Deng, Y., & Rush, A. (2018). Bottom-Up Abstractive Summarization. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 4098-4109). Brussels: Association for Computational Linguistics.

Gehrmann, S., Layne, S., & Dernoncourt, F. (2019). Improving Human Text Comprehension through Semi-Markov CRF-based Neural Section Title Generation. *arXiv*, arXiv:1904.07142v1.

Gidiotis, A., & Tsoumakis, G. (2019). Structured Summarization of Academic Publications. *arXiv*.

Goldberg, Y. (2016). A Primer on Neural Network Models for Natural Language Processing. *Journal of Artificial Intelligence Research, 57*, 345-420.

Goldberg, Y. (2017). *Neural Network Models for Natural Language Processing: Synthesis Lectures on Human Language Technologies.* (G. Hirst, Ed.) Toronto: Morgan & Claypool.

Guo, H., Pasunuru, R., & Bansal, M. (2018). Soft Layer-Specific Multi-Task Summarization with Entailment and Question Generation. *ArXiv*, abs/1805.11004.

Gupta, V., & M., G. (2016). Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review., 47*.

Haghigi, A., & Vanderwende, L. (2009). Exploring content models for multi-document summarization. *Proceedings of the North American Chapter of the Association for Computational Linguistics.* NAACL.

Hardy, & Vlachos, A. (2018). Guided Neural Language Generation for Abstractive Summarization using Abstract Meaning Representation. *arXiv*, arXiv:1808.09160v1.

Hoang, A., Bosselut, A., Celikyilmaz, A., & Choi, Y. (2019). Efficient Adaptation of Pretrained Transformers for Abstractive Summarization. *arXiv*, arXiv:1906.00138v1.

Horev, R. (2018, November 10). *BERT Explained: State of the art language model for NLP*. Retrieved July 2, 2019, from Towards Data Science: https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

Hua, Y., & Wang, L. (2017). A Pilot Study of Domain Adaptation Effect for Neural Abstractive Summarization. *Proceedings of the Workshop on New Frontiers in Summarization* (pp. 100-106). Copenhagen: Association for Computational Linguistics.

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of Tricks for Efficient Text Classification. *arXiv*, arXiv:1607.01759v3.

Joulin, A., Grave, E., Bojanowski, Piotr, Douze, M., Jegou, H., et al. (2016). FASTTEXT.ZIP:COMPRESSING TEXT CLASSIFICATION MODELS. *arXiv*, arXiv:1612.03651v1.

Jurafsky, D., & Martin, J. (2018). N-gram Language Models. In D. Jurafsky, & J. Martin, *Speech and Language Processing* (pp. 37-61). Stanford: Stanford University.

Karn, S.-K., Chen, F., Chen, Y.-Y., Waltinger, U., & Schütze, H. (2019). Generating Multi-Sentence Abstractive Summaries of Interleaved Texts. *arXiv*, arXiv:1906.01973 [.

Keneshloo, Y., Ramakrishnan, N., & Reddy, C. (2018). Deep Transfer Reinforcement Learning for Text Summarization. *ArXiv*, ArXiv:1810.06667v1.

Khan, R., Qian, Y., & Naeem, S. (2019). Extractive based Text Summarization Using K-Means and TF-IDF. *I.J. Information Engineering and Electronic Business*, 33-44.

Khandelwal, U., Clark, K., Jurafsky, D., & Kaiser, L. (2019). Sample Efficient Text Summarization Using a Single Pre-Trained Transformer. *arXiv*, arXiv:1905.08836v1.

Kim, B., Kim, H., & Kim, G. (2019). Abstractive Summarization of Reddit Posts. *arXiv*, arXiv:1811.00783v2.

Kodaira, T., & Komachi, M. (2018). The Rule of Three: Abstractive Text Summarization. *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation.* Hong Kong: Association for Computational Linguistics.

Kryściński, W., Paulus, R., Xiong, C., & Socher, R. (2018). Improving Abstraction in Text Summarization. *Conference: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 1808-1817). CoRR.

Lebanoff, L. e. (2019). Scoring Sentence Singletons and Pairs for Abstractive Summarization. *arXiv*, arXiv:1906.00077v1.

Li, P., Bing, L., & Lam, W. (2018). Actor-Critic based Training Framework for Abstractive Summarization. *CoRR*, 1803.11070.

Li, P., Lam, W., Bing, L., & Wang, Z. (2017). Deep Recurrent Generative Decder for Abstractive Text Summarization. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (p. Association for Computational Linguistics). Copenhagen: Association for Computational Linguistics.

Liao, K., Lebanoff, L., & Liu, F. (2018). Abstract Meaning Representation for Multi-Document Summarization. *The 27th International Conference on Computational Linguistics.* Santa Fe: COLING 2018.

Linqing, L., Yao, L., Min, Y., Qiang, Q., Jia, Z., & Hongyan, L. (2017). Generative Adversarial Network for Abstractive Text Summarization. *arXiv*, arXiv:1711.09357.

Liu, J., Cheung, J., & Louis, A. (2019). What comes next? Extractive summarization by next-sentence prediction. *arXiv*, arXiv:1901.03859v1.

Liu, Y. (2019). Fine-tune BERT for Extractive Summarization. *arXiv*, arXiv:1903.10318v1.

Liu, Y., Titov, I., & Lapata, M. (2019). Single Document Summarization as Tree Induction. *Proceedings of NAACLS-HLT* (pp. 1745-1755). Minneapolis: ACL.

Ljungberg, B. (2017). *Dimensionality reduction for bag-of-words models: PCA vs LSA.* Retrieved July 2019, from Stanford.edu : http://cs229.stanford.edu/proj2017/final-reports/5163902.pdf

Luhn, H. (1958). The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development, 2*(2), 159-165.

Luong, M.-T., Pham, H., & Manning, C. (2015). Effective Approaches to Attention-based Neural Machine Translation. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing,* (pp. 1412-1421). The Association for Computational Linguistics.

MacAvaney, S., Sotudeh, S., Talati, I., Cohen, A., Goharian, N., & Filice, R. (2019). Ontology-Aware Clinical Abstractive Summarization. *srXiv*, arXiv:1905.05818v1.

McClelland, J. L. (2015, December). *The Simple Recurrent Network: A Simple Model that Captures the Structure in Sequences.* Retrieved June 20, 2019, from Stanford web: https://web.stanford.edu/group/pdplab/pdphandbook/handbookch8.html

Mehta, P., Aurora, G., & Majumder, P. (2018). Attention based Sentence Extraction from Scientific Articles using Pseudo-Labeled data. *CoRR*.

Mendoza, V. N., Ledeneva, Y., & García-Hernández, R. A. (2019). Abstractive Multi-Document Text Summarization Using a Genetic Algorithm. *11th Mexican Conference,MCPR 2019: Pattern Recognition* (pp. 422-432). Springer.

Mihalcea, R. (2004). TextRank:Bringing Order into Texts. *ACL*.

Mikolov, T., Corrado, G., Chen, K., & Dean, J. (2013b). Efficient Estimation of Word Representations in Vector Space. *International Conference on Learning Representations.*

Mikolov, T., Sutskever, I., K, C., Corrado, G., & Dean, J. (2013a). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 3111-3119.

Miller, D. (2019). Leveraging BERT for Extractive Text Summarization on Lectures. *arXiv*, arXiv:1906.04165 .

Moroshko, E., Feigenblat, G., Roitman, H., & Konopicki, D. (2019). An Editorial Network for Enhanced Document Summarization. *arXiv*, arXiv:1902.10360v1.

Nallapati, R., Xiang, B., & Zhou, B. (2016). Abstractive text summarization using sequence-to-sequence rnns and beyond. *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning* (pp. 280-290). CoNLL.

Nallapati, R., Zhai, F., & Zhou, B. (2017). SummaRuNNer: A Recurrent Neural Network based Sequence Model for Extractive Summarization of Documents. *Thirty-First AAAI Conference on Artificial Intelligence* (pp. 3075-3081). AAAI Press.

Narayan, S., Cohen, S., & Lapata, M. (2018). Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. *Proceedings of the Conference on Empirical Methods in Natural Language Processing.* EMNLP.

Narayan, S., Cohen, S., & Lapata, M. (2018). Ranking Sentences for Extractive Summarization with Reinforcement Learning. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (pp. 1747-1759). Association for Computational Linguistics.

Narayan, S., Papasarantopoulos, N., Lapata, M., & Cohen, S. (2017). Neural Extractive Summarization with Side Information. *CoRR*, 1704.04530.

Nayeem, M., Fuad, T., & Chali, Y. (2019). Neural Diverse Abstractive Sentence Compression Generation. *Lecture Notes in Computer Science. 11438*, pp. 109-116. Cologne: Springer.

Nema, P., Khapra, M., Laha, A., & Ravindran, B. (2017). Diversity driven Attention Model for Query-based Abstractive Summarization. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (pp. 1063-1072). Association for Computational Linguistics (ACL).

Nenkova, A., & Passonneau, R. (2004). Evaluating Content Selection in Summarization: The Pyramid Methof. *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004* (pp. 145-152). Boston: Association for Computational Linguistics.

Nenkova, A., R, P., & McKeown, K. (2007). The Pyramid Method: Incorporating human content selection variation in summarization evaluation. *ACM Transactions on Speech and Language Processing (TSLP), 4*(7).

Omar, N., & Duru, N. (2017). Text Summarization and Evaluation Methods- An Overview. *International Journal of Engineering Research and Application, 7*(12), 89-93.

Ouyang, J., Song, B., & McKeown, K. (2019). A Robust Abstractive System for Cross-Lingual Summarization. *Proceedings of the 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* Minneapolis: NAACL-HLT'19.

Palaskar, S., Libovicky, J., Gella, S., & Metze, F. (2019). Multimodal Abstractive Summarization for How2 Videos. *arXiv,* arXiv:1906.07901v1.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 311-318). Philadelphia: ACL.

Parmar, C., Chaubey, R., Bhatt, K., & Lokare, R. (2019). Abstractive Text Summarization using Artificial Intelligence. *2nd International Conference on Advances in Science and Technology (ICAST-2019).* Mumbai: K J Somaiya Institute of Engineering & Information Technology.

Pasunuru, R., & Bansal, M. (2018). Multi-Reward Reinforced Summarization with Saliency and Entailment. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers).* New Orleans: Association for Computational Linguistics.

Paulus, R., Xiong, C., & Socher, R. (2017). A deep reinforced model for abstractive summarization. *International Conference on Learning Representations.*

Pennington, J., Socher, R., & Manning, C. (2014). GloVe:Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (pp. 1532-1543). Doha: Association for Computational Linguistics.

Peters, M. e. (2018). Deep contextualized word representations. *Proceedings of NAACL-HLT* (pp. 2227-2237). New Orleans: Association for Computational Linguistics.

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

Rarnpradab, S., Liu, F., & Hua, K. A. (2017). Toward Extractive Summarization of Online Forum Discussions via Hierarchical Attention Networkds. *Proceedings of the Thirtieth International Florida Artificial Intelligence Research Society Conference.* AAAI.

Raschka, S. (2014, August 3rd). *https://sebastianraschka.com/Articles/2014_python_lda.html*. Retrieved July 1st, 2019, from Sebastian Raschka: https://sebastianraschka.com/Articles/2014_python_lda.html

Ratia, T. (2018, July 17). *Frase*. Retrieved May 13th, 2019, from 20 Applications of Automatic Summarization in the Enterprise: https://blog.frase.io/20-applications-of-automatic-summarization-in-the-enterprise/

Reiter, E., & Dale, R. (1997). Building Applied Natural Language Generation Systems. *Natural Language Engineering, 1*(1).

Rennie, S., Marcheret, E., Mroueh, Y., Ross, J., & Goel, V. (2016). Self-critical sequence training for image captioning. *arXiv*, arXiv: 1612.00563 .

Saggion, H., & Poibeau, T. (2013). Automatic Text Summarization: Past, Present and Future. In T. e. Poibeau, *Multi-source, Multilingual Information Extraction and Summarization* (pp. 3-21). Springer.

See, A., Liu, P., & Manning, C. (2017). Get to the Point: Summarization with Pointer-Generator Networks. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics.* Vancouver.

Singh, A., & Shashi, M. (2019). Deep Learning Architecture for Multi-Document Summarization as a cascade of Abstractive and Extractive Summarization approaches. *International Journal of Computer Sciences and Engineering*, DOI: 10.26438/ijcse/v7i3.950954.

Sinha, A., Yadav, A., & Gahlot, A. (2018). Extractive Text Summarization using Neural Networks. *ArXiv*, abs/1802.10137.

Song, K., Zhao, L., & Liu, F. (2018). Structure-Infused Copy Mechanisms for Abstractive Summarization. *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 1717-1729). Santa Fe: Association for Computational Linguistics.

Song, S., Huang, H., & Ruan, T. (2018). Abstractive text summarization using LSTM-CNN based deep learning. *Multimedia Tools and Applications. 78*, pp. 857–875. 2018: Springer.

Steinberger, J., & Ježek, K. (2009). Evaluation measures for Text Summarization. *Computing and Informatics, 28*, 1001-1026.

Tan, J., Wan, X., & Xiao, J. (2017). Abstractive doctument summarization with a graph-based attentional neural network. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017* (pp. 1171-1181). Vancouver: Volume 1: Long Papers Association for Computational Linguistics.

Vanderwernde, L. e. (2007). Beyond SumBasic: Task focused summarization with sentence simplification and lexical expansion. *Information processing and Managment, 43*(6), 1606-1618.

Verma, R., & Daniel, L. (2017). Extractive Summarization: Limits, Compression, Generalized Model and Heuristics. *Computacion y Sistemas, 21*(4).

Wang, C.-F. (2019, January 8). *The Vanishing Gradient Problem*. Retrieved June 2019, from Towards data science: https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484

Wong, K., Wu, M., & W., L. (2008). Extractive Summarization using Supervised and Semi-supervised Learning. *COLING '08 Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, (pp. 985-992). Manchester.

Wu, Y., & Hu, B. (2018). Learning to Extract Coherent Summary via Deep Reinforcement Learning. *CoRR*, abs/1804.07036.

Xie, N., Li, S., Ren, H., & Zhai, Q. (2018). Abstractive Summarization Improved by WordNet-based Extractive Sentences. *CCF International Conference on Natural Language Processing and Chinese Computing* (pp. 404-415). NLPCC 2018.

Xu, J., & Durrett, G. (2019). Neural Extractive Text Summarization with Syntactic Compression. *arXiv*, arXiv:1902.00863v1.

Xu, W., Napoles, C., Chen, Q., Pavlick, E., & Chris, B.-C. (2016). Optimizing Statistical Machine Translation for Text Simplification. *Transactions of the Association for Computational Linguistics, Volume 4* (pp. 401-415). ACL.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, A. (2016). Hierarchical attention networkds for document classification. *In Proceedings of the North American Chapter of the Association for Computational Linguistics.* NAACL.

Young, T., Harazika, D., Poria, S., & Cambria, E. (2018, August). Recent Trends in Deep Learning Based Natural Language Processing. *IEEE Computational Intelligence Magazine, 13*(3), pp. 55-75.

Zaremba, W., & Sutskever, I. (2015). Reinforcement learninf neural turing machines. *arXiv*, preprint arXiv:1505.00521 362.

Zhang, H., Xu, J., & Wang, J. (2019). Pretraining-Based Natural Language Generation for Text Summarization. *arXiv*, arXiv:1902.09243v2.

Zhang, J., Tan, J., & Wan, X. (2018). Towards a Neural Network Approach to Abstractive Multi-Document Summarization. *arXiv*, 1804.09010v1.

Zhang, X., Lapata, M., Wei, F., & Zhou, M. (2018). Neural Latent Extractive Document Summarization. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 779-784). Association for Computational Linguistics.

Zhang, Y., Li, D., Wang, Y., Fang, Y., & Xiao, W. (2019). Abstract Text Summarization with a Convolutional Seq2Seq Model. *Applied Sciences 9(8):1665*.

Zhou, Q., Yang, N., Wei, F., Huang, S., Zhou, M., & Zhao, T. (2018). Neural Document Summarization by Jointly Learning to Score and Select Sentences. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 654-663). Melbourne: Association for Computational Linguistics.