Diedrichsen, Elke (2017) Linguistic challenges in automatic summarization technology. Journal of Computer-Assisted Linguistic Research 1: 40-60.

Linguistic challenges in automatic summarization technology

Elke Diedrichsen
Computational and Functional Linguistics Research Group
Institute of Technology Blanchardstown, Dublin, Ireland
e.diedric@googlemail.com

Received: 26 February 2017 / Accepted: 30 April 2017 / Published: 19 June 2017

Abstract
Automatic summarization is a field of Natural Language Processing that is increasingly used in industry today. The goal of the summarization process is to create a summary of one document or a multiplicity of documents that will retain the sense and the most important aspects while reducing the length considerably, to a size that may be user-defined. One differentiates between extraction-based and abstraction-based summarization. In an extraction-based system, the words and sentences are copied out of the original source without any modification. An abstraction-based summary can compress, fuse or paraphrase sections of the source document. As of today, most summarization systems are extractive. Automatic document summarization technology presents interesting challenges for Natural Language Processing. It works on the basis of coreference resolution, discourse analysis, named entity recognition (NER), information extraction (IE), natural language understanding, topic segmentation and recognition, word segmentation and part-of-speech tagging. This study will overview some current approaches to the implementation of auto summarization technology and discuss the state of the art of the most important NLP tasks involved in them. We will pay particular attention to current methods of sentence extraction and compression for single and multi-document summarization, as these applications are based on theories of syntax and discourse and their implementation therefore requires a solid background in linguistics. Summarization technologies are also used for image collection summarization and video summarization, but the scope of this paper will be limited to document summarization.

Keywords: automatic summarization, natural language processing, linguistics, syntax, discourse

DOI: https://doi.org/10.4995/jclr.2017.7787
PUBLISHER: UNIVERSITAT POLITÈCNICA DE VALÈNCIA.

1. Introduction

The automatic summarization of documents has been in use since the middle of the last century, and it has been developed and adapted to new technology ever since. In particular, the recent availability of the World Wide Web as an information resource has called for means to create extracts of the wealth of available information – be it user query specific or general.

There are various kinds of summaries. One distinguishes extractive summaries from abstractive summaries. This distinction relates to the quality of the produced summary. An extractive summary is essentially a shrunk version of the original, as it extracts sentences from the source, but replicates them with the same words. An abstractive summary is more sophisticated, as it conveys the main information without quoting literally from the original.

Furthermore, one distinguishes between single and multi-document summarization. Multi-document summarization is specifically useful for information extraction from the web. Several documents that relate to one given topic can be summarized.

There is another classification of automatic summaries that is based on content. An indicative summary is one that informs the reader what a given text or set of texts is about. An informative summary, on the other hand, is one that reproduces the main information of the original and can be used as a replacement of it (Nencova and McKeown 2011; Jurafsky and Martin 2009). As for the methods currently used for creating automatic summaries, it is the state of the art that most of the systems produce extractive summaries and use sentence extraction. The challenge is for the system to determine which sentences are important enough to comprise a summary of the full text and retain the main information. There have been recent advances in abstractive summarization as well, which involve technologies that enable sentence compression and fusion, processing of context and discourse reference tracking, semantic interpretation and automatic generation of summary language (Nencova and McKeown 2011: 108).

This chapter will focus on linguistic challenges posed by methods of auto-summarization. To date, the most prevalent method is the extractive summarization of one or multiple documents. In section 2, we will introduce major methods and challenges of extracting sentences for various applications. These include information retrieval, determination of importance, achieving order and coherence and avoiding redundancy and repetition. Section 3 will introduce sentence compression, which is an abstractive method. Sentence compression is used in order to create a shorter version of the text that includes not only fewer, but shorter sentences. In order to achieve this, the main information out of each sentence has to be extracted. It will be shown that sentence compression requires syntactic parsing of the input sentences, and it is therefore a particularly linguistic task. Furthermore, there are methods of sentence compression that take into account discourse and context information in order to achieve coherence. Section 4 will give a short discussion of the issue of summarization evaluation, which is a particularly challenging task as well because the quality of the summarization may vary from one recipient to the next. Section 4 will also give a summary and outlook of the issues discussed in this chapter.

2. Sentence Extraction: Methods and challenges

The state of the art with auto-summarization technology is that extractive summarization still by far outweighs abstractive summarization with respect to the available systems on the market. Even though abstractive summaries represent the more sophisticated approach, as their outputs are designed to resemble human summaries more closely, they are still not advanced enough to outperform extractive summaries from an evaluation perspective (Nencova and McKeown 2011: 156).

This section deals with the challenges that sentence extraction poses, and it will discuss some methods that have been found to overcome the challenges. The next section will introduce abstractive summarization, where sentences are compressed and fused in order to arrive at a more concise, topic-oriented summary result.

Generally, in a sentence extraction approach, the summary is created by the selection of sentences from the original document. The sentences are not altered or re-ordered, so the main challenge is to find the sentences that provide the most important information, i.e. the information that needs to be included in order for the resulting summary to cover the main thematic information from the source text. An extractive summary like this is likely to contain redundant or less important pieces of text, as full sentences are extracted as they are. It is also at risk of being repetitive, especially if it is a multi-document summary, which summarizes several documents that deal with the same topic. Also, the resulting summary may have deficits regarding its readability, as it does not use alternative wording and sentence conjunction, which are features often found in summaries created by a human.

The following subsections discuss approaches to sentence extraction technology. The first and most widespread one, which is completely data-driven and therefore does not rely on human annotation at all, covers features for determining the importance of a sentence. These are generally centered around tracking frequency, redundancy and repetition (Nencova and McKeown 2011: 120).

2.1. Word probability for content words

For sentence extraction, good results have been achieved by the use of features that are entirely independent of any human intervention. They do not need any human-made models or any linguistic processing, and they do not require interpretation or evaluation from external knowledge sources (Nencova and McKeown 2011: 121-125).

As a first step, frequency is taken as an indicator of sentence importance. An early approach by Luhn (1958) started out from the idea that lexemes occurring frequently in a piece of text would be an indicator of things that are very important and topical. However, function words like determiners, pronouns, conjunctions and the like would disturb this picture, as they will naturally appear frequently in any text. So, how can the topical words be filtered out automatically?

In this simple approach to determining importance on the basis of frequency, the probability of a word w, which is the value p(w), is calculated as the number of times this word appears in the text, c(w), divided by the number of all words in the input, N (Nencova and McKeown 2011: 122).

(1) $P(w) = c(w) / N$

The likelihood of a summary can be computed on the basis of a multinomial distribution. For the calculation, M will be the number of words in the summary, the factorial of which will be divided by the factorial of the number of times the word appears in the summary and multiplied by the probability of the appearance of that word in the summary, as estimated from the input documents. Another system, SumBasic, works through the sentences in a greedy fashion, which means that it proceeds sentence by sentence and selects the sentence which contains the word that has the highest probability so far. The approach assumes that the word with the highest probability represents the most important topic, and the goal is to select the sentence that contains this word. The greedy selection works as a loop that will be repeated until the required length of the summary is reached. The greedy, sentence by sentence procedure of selection is, however, outperformed by a global approach that optimizes the occurrence of important words over the entire text (see also Yih, Goodman, Vanderwende and Suzuki 2007).

2.2. TF*IDF weights

If one considers frequency as the main indicator of importance, the problem is that in any text, certain words will occur frequently without being indicative of topicality. In Information Retrieval, the most frequent words will be collected in a stop word list. The stop word list will, depending on the language, contain function words like determiners, pronouns, prepositions and the like, and among content words, it will contain common domain words. Determining a stop word list across documents may not lead to the desired results, however, and therefore another method from Information Retrieval, the TF*IDF weighting of words, can be used. TF*IDF stands for Term Frequency*Inverse Document Frequency. This method accounts for the fact that the mere frequency of a word may be an indicator of its topicality, but it could also be an indicator of it being a stop word. So how does one pick out the topical words on the basis of frequency, and simultaneously avoid ending up with a mixture of topical words and stop words? In order to calculate the weight of a word w, its frequency in a large background corpus is measured. The background corpus usually contains texts from a similar genre as the input document, which is the document to be summarized. Like this, the frequency of appearance of words pertaining to a given genre is obtained.

For the calculation of the weight of a word, where weight means essentially the extent to which it is burdened with topical information, the following values are required: its frequency in the input document c(w), on the one hand, and the number of documents d(w) from a background corpus D that contain the word, on the other hand.

The inverse document frequency IDF is calculated through the following formula:

(2) $TF*IDFw = c(w) \times \log D/d(w)$

This method is easy and fast to compute, and it arrives at the desired results, as it accounts for the following facts: Descriptive topic words for a topic area are very frequent in documents from one topic area, but they would not appear frequently in other documents. Stop words, on the other hand, will be frequent in any document. The calculation described above rules them out, as their IDF weight will be close to zero.

Because of its good results and easy application, TF*IDF is one of the most widely used features

that is part of most extractive summarization applications to date (Nencova and McKeown 2011: 124-125; also Filatova and Hatzivassiloglou 2004).

2.3. Log-likelihood ratio test for topic signatures

This test is described as more powerful than the previously described method by Nencova and McKeown (2011: 125-128). It extends the potential of the TF*IDF calculation in that it not only figures out the highly descriptive, topical words, which have been called *topic signatures* in the literature (Lin and Hovy 2000), but also provides a threshold to determine which words and sentences are descriptive enough to be included in the summary, and which are not.

This is a statistical method that works by comparing the likelihood of a word to appear in an input corpus versus is likelihood to appear in a background corpus. The determination of the topic signatures, which are the topical words, is achieved by measuring the statistical significance of their probability: If the value is above the value for chance distribution on a $\chi 2$ distribution table, the word can count as a topic signature word.

> Topic signature words are those that have a likelihood statistic greater than what one would expect by chance (Nencova and McKeown 2011: 126).

This approach works without assigning weight to single words. Instead, it assigns importance values to sentences. The importance of a sentence is measured in terms of the number of topic signatures it contains. Research has shown that the identification of topic signature terms achieves good results in summarization evaluation. It produces automatic summaries that are similar to summaries produced by a human with respect to the coverage of topical information. This approach has been used both for single and for multiple document summarization.

2.4. Sentence clustering in multi-document summarization

In multi-document summarization, the input for the summary comprises several sentences. Multi-document summarization is applied when a variety of sources is used in order to obtain information about one or more topics. The typical use would be a scan of several news articles with respect to a topic of interest. How is the most important information filtered out when there are several sources with topical information?

As a first step, the importance of the information is assigned on the basis of frequency again, by checking which information occurs in many or all of the input documents. Similar sentences from the source documents are clustered together (Nencova and McKeown 2011: 128; Hatzivassiloglou, Klavans, Holcombe, Barzilay, yen Kan, McKeown 2001; Siddhartan, Nenkova, McKeown 2004). If a cluster has many sentences in it, this is taken as an indicator that it represents an important topic from the input. In order to obtain a succinct and generally non-redundant extractive summary of the input, one sentence is selected from each main cluster. This approach works on sentence level rather than on word level, and each sentence is assigned to only one cluster. Redundancy and lack of precision may occur when there are many sentences that cover more than one topic. The approach discussed in the next subsection is able to account for this potential problem.

2.5. Graph-based methods for sentence ranking

Approaches using graphs combine the advantages from word frequency and sentence clustering methods discussed earlier (Nencova and McKeown 2011: 128-130; Qazvinian and Radev 2008). These approaches work both on the word level and on the sentence level: Sentence similarity is measured on the basis of word overlap. Words that occur frequently will therefore be found in many sentences, and sentences that are similar with respect to the words they contain will be considered as important. Like in the topic word approach, the focus of the method is to figure out the most important content. There is generally no ordering of sentences. The approach can be used cross-linguistically, as language-specific linguistic processing is not necessary. However, if linguistic information like syntax and semantic roles is incorporated, the results can be improved. Graph-based approaches have been found to work for both single and multiple document summarization.

2.6. Methods using machine learning

Supervised methods using machine learning make use of the fact that there are many factors that indicate sentence importance. With these methods, developers are generally trying to incorporate more than one feature. In machine learning, training corpora with existing summaries are matched against an input corpus. Many recent machine learning approaches use Hidden Markov Models (HMM). Generally, the machine learning approaches have not been able to outperform unsupervised, data driven models. However, as far as domain or genre specific summarization is concerned, machine learning approaches are more successful. Their classifiers are trained to identify specified information in relation to the kind of text that the summarizer is supposed to work on (Nencova and McKeown 2011: 132-134).

Machine learning approaches are more complicated and time consuming. In order to provide training corpora for the system, human annotators would be required to select sentences that they consider to be important enough to go into the summary. There are methods that align human summaries and input summaries, in order to provide labelled data to train their systems. The problem here is, however, that the selection of summary-worthy information may be subjective to a certain degree, such that more than one training summary would be required to find the range of potentially important sentences. The state of the art is that so far, for generic summarization tasks, machine learning approaches in all their complexity are not good enough to outperform simple unsupervised methods using a single feature.

3. Abstractive Summarization: Advances in sentence compression based on syntax and discourse

Extractive summaries are widely used, and there is extensive research in that area. Extractive summarization is mainly about finding the sentences which carry the most important and therefore, most summary-worthy information. As we have seen, non-supervised and data-driven methods for sentence extraction have turned out to be very successful in fulfilling the tasks demanded of them.

However, extractive summaries have many drawbacks, especially when compared to human-made summaries. As extractive summaries look for full sentences including important information, they reproduce the sentences fully, and do not account for linguistic variation, readability, and they do not remove redundant parts from the selected sentences. According to Nencova and McKeown (2011: 152), there are only few advances in the field of abstractive summarization.

This section will discuss sentence compression. It will be shown that sentence compression relies on linguistic analysis more than the extractive approaches introduced earlier. Sentence compression is a method that essentially shortens the selected sentences by removing material that is not needed for the summary. In order to account for readability and coherence, methods have been found that make context-dependent revisions with respect to the choice of referential expressions, and that improve the information ordering in the output summaries. These are discussed here as well. Research has shown that in order to create summaries that align as closely as possible with human-made summaries, one step is to extract the main information from the input as a whole by selecting sentences that contain the information. Depending on the input, however, those selected sentences may be long and complex, and they may contain material that is not necessary for the summary. Human summarizers would therefore always remove unnecessary bits from sentences themselves and reformulate them to arrive at a fluent new text. Doing this automatically is certainly a greater challenge than the one posed by merely extracting the sentences.

There are rule-based, linguistic approaches to sentence compression, and there are approaches that are mainly based on statistics. The following subsections will concentrate on the methods that involve linguistics, as they are of most concern for this volume.

3.1. Automatic sentence compression

For automatic sentence compression, the implementation of linguistic rules involves both syntactic and discourse knowledge. Some very promising work initiated by Jing (2000; 2001) and Jing and McKeown (2000) uses syntactic information to determine importance within the sentence, and discourse information is used in order to link the contents of the resulting sentence with the rest of the summary.

In a syntactic approach that uses tree structures, all sentences that are candidates for the summary are parsed, and all nodes in the tree that are needed to retain the grammaticality of the sentence are marked. The marking alerts the reduction module that these components cannot be removed. Those components will be the main verb, the head of a noun phrase, and the obligatory arguments of the verb.

Note that in Role and Reference Grammar (RRG) (Van Valin 2005), a non-generative approach to this syntactic selection procedure, the CORE of a sentence would most probably be the part selected as the most important centre of information, as it represents the minimal syntactic structure of a sentence with the main verb and its obligatory arguments. RRG is in many respects superior to generative multistratal tree structure approaches to syntax, as contrary to those, RRG does not assume an abstract deep structure for semantics, but represents a theory that links semantic roles and basic discourse functions directly into the syntactic structure. RRG also provides a solid conception of the parts of the theory that are language specific, and those that are valid cross-linguistically. These are all qualities that cannot be found in generative approaches to syntax, as these are generally based on findings in English alone.

It would be an interesting theoretical challenge to come up with an RRG-based approach to automatic summarization. Currently, RRG is starting to be applied to the challenges of NLP. There is, for example, a linguistic parser for basic German sentences that shows promising results as a proof-of-concept (Diedrichsen 2014). See also Nolan and Periñan-Pascual (2014) for other NLP applications using RRG.

In Jing's original approach for automatic sentence compression (2001), contextual information is used in addition to the syntactic information, in order to figure out the parts that are linked to the overall topic of the input. These are also marked for non-deletion, even if they are syntactically not central. Jing also uses a statistical method to compute the likelihood of a human deleting a type of constituent by use of a corpus of human-made professional summaries (see also Nencova and McKeown 2011: 155).

3.2. Trimmer

In an approach by Zajic, Dorr, Lin and Schwartz (2007) the Trimmer system employs a linguistically-motivated algorithm to trim syntactic constituents from sentences, until the required summary length has been reached. The summary will have the syntactic form of a headline, that provides the main information from the text in a very compressed format. Topiary is a variant that includes topic terms. HMM edge is a noisy-channel approach that starts out from the language typical for headlines and removes all *noise* from the text that appears around the central headline-worthy information. We will now discuss some linguistically based approaches to sentence compression.

Trimmer is based on studies that have shown that in successful human summaries, certain sentence constituents are retained for a summary of a given text, while others are generally removed. Trimmer uses the output of a constituency parser that uses the Penn Treebank conventions. The approach involves removing grammatical constituents according to a set of rules, which are applied iteratively to a sentence until a length threshold has been reached (Zajic, Dorr, Lin and Schwartz 2007: 1552-1553). The resulting sentences generally have the style and quality of headlines.

The Trimmer Algorithm uses the following rules (see Zajic, Dorr, Lin and Schwartz 2007: 1553):

(3) The Trimmer Algorithm

1. Remove temporal expressions

2. Select Root S node
3. Remove preposed adjuncts
4. Remove some determiners
5. Remove conjunctions
6. Remove modal verbs
7. Remove complementizer that
8. Apply the XP over XP rule
9. Remove PPs that do not contain named entities
10. Remove all PPs under SBARs
11. Remove SBARs
12. Backtrack to state before Step 9
13. Remove SBARs
14. Remove PPs that do not contain named entities
15. Remove all PPs

Temporal expressions and determiners are generally considered to be units with low content, and therefore not important for the summary. The removal of determiners leads directly to the headline style of the resulting summary. Step 2 involves choosing the lowest leftmost Root S node as the root node of the headline. A Root S node is labelled S in the parse tree. Its children, i.e. sub-roots, are labelled NP and VP, and they appear in that order. The ordering constraints make it apparent that this parser and trimming algorithm is oriented at the English language and its word order conventions. The Generative grammar framework that is the basis of it accounts for these ordering constraints. An approach like this would lead to difficulties in languages where the word order is more flexible. RRG is superior here, as it does not depend on word order, and it does not commit to the notion of a VP. Research has shown that many languages do not have a VP. The concept of a VP is based on English. It is a theory-internal concept of Generative Grammar.

Step 3 is also based on a word order matter: Preposed adjuncts are removed. Preposed adjuncts are mostly realised as preambles of a sentence, as in *According to xy* or *it is believed that...*

From Step 5 downwards, the steps remove peripheral material by iteratively deleting constituents until the length threshold is reached. Each rule is successively applied to the "…deepest, rightmost remaining node in the pool" (Zajic, Dorr, Lin and Schwartz 2007: 1555). Each step is a rule that goes through all possible nodes in the parse tree, before the next step is initiated.

The XP-over-XP rule in Step 8 removes embedded information in sentences and noun phrases. XP as a variable can have the value NP or VP. Therefore, the rule covers embedding and recursion

in the NP and VP domains. It states that immediate child constituents of a higher XP can be removed. The rule therefore removes relative clauses, for example, and appositives.

Steps 9 through 15 remove prepositional phrases and subordinate clauses. These are applied at a late stage in the algorithm, when all other rules have been applied, in order to ensure that they do not remove important content (Zajic, Dorr, Lin and Schwartz 2007: 1554-1555). The algorithm provides a back-off option for them as well. With prepositional phrases, there is a certain danger that they would contain important information that could be needed for the summary. On the other hand, they represent smaller units than subordinate clauses, so the system has a trial and a back-off mechanism for the removal of these entities: First the PP rule is applied (Step 9 and 10), but if this does not lead to the desired length, the system reverts to the state before this last step and removes subordinate clauses by application of the SBAR rule. If this still does not lead to the threshold length, the PP rule is applied again.

In order to avoid removing PPs that contain important information, an application called IdentiFinder by BBN is used that distinguishes between PPs containing important named entities and those containing only temporal expressions (Zajic, Dorr, Lin and Schwartz 2007: 1556). IdentiFinder marks and removes temporal expressions. It also marks PPs that represent named entities and ensures that the removal of PPs spares PPs with named entities, as they may contain very important locational information.

3.3. Topiary: Topic term generation and event description

Trimmer is a potent algorithm to extract the important information from a sentence by automatically applying linguistic rules. Its potential drawbacks include the restricted applicability – using a generative, linearization-based grammar model, it will be safely applicable only to English. Also, the headline-esque output lacks fluency and is not embedded in a context.

The authors themselves remark that the headline output generated from single sentences may not suffice to cover the important information from the input text, and the syntactically generated headline may miss topical information. Therefore, it is concluded that it would be good to include topic information in a summarization application. Some approaches tackle this by constructing headlines from lists of topic terms (Lewis 1999; Schwartz, Imai, Kubala, Makhoul 1997). For example, Unsupervised Topic Discovery (UTD) (Zajic, Dorr, Schwartz 2004; Sista, Schwartz, Leek, Makhoul 2002) uses as input an unannotated corpus and creates a set of topic models with meaningful names (Zajic, Dorr, Lin and Schwartz 2007: 1556-1557).

UTD works in several stages. It analyses the corpus to find frequently occurring strings of words. These tend to be meaningful names of topics. Using a modified TF*IDF measure, it finds the topical phrases in each document. These are considered to be an initial set of topic names. They provide topic names for each of the documents. In a third stage, UTD trains topic models on the basis of these topic names. The topic models are then used to find the most likely topics for each document in a fourth stage (Zajic, Dorr, Schwartz 2004).

Approaches like this, however, have the disadvantage that they do not contain verbs. They indicate the general subject matter, but do not inform about events that happened. Therefore, the Topiary system has been developed which combines Trimmer with approaches that discover

topical information in order to produce a fluent summary with relevant context. Topiary uses UTD to generate topic terms for a multiplicity of input documents, and it combines it with another approach, OnTopic, to assign the topic terms to the documents. Now that a list of topic terms has been generated, they are combined with the sentence compression approach used in the Trimmer system.

By separating out the topic selection process and the trimming of relevant sentences, Topiary accounts for the fact that the topical information and the compressed sentence may not occur together in the source text, and it therefore provides a concise summary of the matter that captures the most central information without relying on the topicality of singular input sentences. It also retains a natural flow for the summary by providing full sentences with verbs along with the topic list. It achieves this by first applying Trimmer rules to the sentences, so that each sentence is stored in its compressed version. A list of topic names with relevance scores is taken as additional input. Starting out from the longest one of the compressed variants, the compression threshold is dynamically lowered so there is room to select more topic words, which are not yet in the headline. After the trimming is complete, the system adds additional topic terms that do not occur in the headline, until all space is used up (Zajic, Dorr, Schwartz 2004; Zajic, Dorr, Lin and Schwartz 2007: 1557-1558).

For example, with the UTD output in (4), the following Trimmer (5) and Topiary (6) outputs are achieved. The story underlying these is about the FBI investigation of the 1998 bombing of the U.S. embassy in Nairobi (Zajic, Dorr, Lin and Schwartz 2007: 1557-1558).

(4) BIN LADEN, EMBASSY, BOMBING, POLICE OFFICIALS, PRISON, HOUSE, FIRE, KABILA

(5) FBI agents this week began questioning relatives of the victims

(6) BIN LADEN, EMBASSY, BOMBING: FBI agents this week began questioning relatives

This combination of a topic list with parse-and-trim compression won Topiary the highest score on the single-document summarization task in the Document Understanding Conference (DUC), an annual workshop for the evaluation of document summarization systems, in 2004.

3.4. HMM Hedge

The approach to sentence compression implemented in HMM Hedge works as if the observed data, which is the input story, were the result of the unobserved data (the headlines) being distorted by transmission through a noisy channel. To achieve this, a unigram model of general language is generated by use of a large corpus of news stories. The process by which the story is generated from a headline is computed via a Hidden Markov Model (HMM). Hidden Markov Models, first described and applied by Baum and others (Baum, Petrie, Soules, Weiss 1970; Baum 1972), are weighted finite-state automatons. A HMM works on situations in which there are unobserved (hidden) states. It is a stochastic signal model that allows one to talk about observed states and hidden states (Jurafsky and Martin 2009: 179; Rabiner 1989). Hidden Markov models are used in many NLP tasks like part-of-speech tagging and speech recognition in order to separate the observable signal (the observed word, the perceived sound) from the hidden

categories (the part-of-speech tag, the word) and build probabilistic models of the occurrences of the hidden states. For generating headlines, a HMM model consists of two states, H and G. H is the state that emits words that occur in the headline, and G is the state that emits all other words in the input story. A H state can only emit the word that it represents, while the corresponding G state remembers which word was emitted by its H state, and it can also emit any other word in the story language. A headline is basically a path through the HMM from the start to the end state that emits all words in the story in the correct order. The HMM will transition between the states H and G in order to generate the words of the story. H states put out only the headline words, so that at the end of the process the whole story is created and the headline words are separated out (Zajic, Dorr, Lin and Schwartz 2007: 1559).

HMM edge has three additional decoding parameters to make sure the system chooses headline output that best resembles actual headlines. These are the position bias, the clump bias and the gap bias. The position bias favours headlines made up of words from the beginning of the story. It has been found that with the exception of human interest and sports stories, which contain opening teasers to attract the readers' attention, the front of the story tends to contain headline-worthy information. The clump bias accounts for the fact that headlines created by humans have a tendency to contain contiguous blocks of topical words. This bias therefore enables the generation of headlines where *clumpiness* of topic-worthy information is pursued. The gap bias has been induced in order to avoid large proportions of non-topical text between the topical *clumps*. This bias penalizes the system for staying in a G state for too long, and makes sure the headline output has only a few large gaps.

In order to arrive at the appropriate headline language called "Headlinese", a morphological variant is incorporated in the H state that creates a change from the past tense often observed in story words into the present tense that is appropriate for headline verbs. Also, there is a constraint on the selection of headlines that require them to contain at least one verb. With HMM Hedge, multiple alternative compressions of a sentence can be obtained (Zajic, Dorr, Lin and Schwartz 2007: 1558-1560).

Example (7) shows the compression of a sentence using HMM Hedge.

(7) (after Zajic, Dorr, Lin and Schwartz 2007: 1560)

a) A group has proposed awarding $1 million in every general election to one randomly chosen voter.

b) Group proposes awarding $1 million to randomly chosen voter.

3.5. CLASSY

The system CLASSY (Clustering, Linguistics, and Statistics for Summarization Yield) represents a combination of syntactic and lexical heuristics to compress sentences (Nencova and McKeown 2011: 156). CLASSY was developed by Conroy and O'Leary (2001), see also Conroy, Schlesinger and Goldstein Stewart (2005) and Conroy, Schlesinger, O'Leary and Goldstein (2006). The summarizer is based on a HMM for single document sentence selection and a pivoted QR algorithm for the generation of a multi-document summary. Later, the system was modified to include linguistic capabilities. The version discussed here is focused on query based methods of

summarization. It is enhanced year by year in order to include its evaluation results and to adapt to different challenges.

For the linguistic technology employed in CLASSY, shallow parsing is used along with part-of-speech tagging in order to exclude phrases and lexical material. Phrase eliminations are applied according to the following categorisation:

1. Gerund clauses
2. Restricted relative-clause appositives
3. Intra-sentential attribution
4. Lead adverbs

These phrase eliminations are applied before summarization. In order to arrive at a query-based summarization, query terms are identified for each document set and the summaries are oriented at the questions asked in the topic descriptions. In addition to that, a named entity identifier, BBN's IdentiFinder is employed in order to help answer queries with respect to named entities like people, locations, and organisations.

The linguistic processing is carried out before the scoring by a Hidden Markov Model (HMM) and selection of the summary sentences by a pivoted QR factorization, which is a rank revealing algorithm, is done. The HMM used in CLASSY has two kinds of states. One applies to summary sentences and the other one to non-summary sentences. The features used to distinguish between those states are based on a stop list and on the query terms (Conroy, Schlesinger and Goldstein Stewart 2005: 3-4).

In a later version of CLASSY presented at DUC 2006, the POS tagger was eliminated from the sentence trimmer and replaced by a new application that did not involve dependence on POS tags, as these had proven to lead to unintended results. The word lists in this later version are created ad hoc for the code and can be adapted depending on the document. The word lists now include function words like prepositions, conjunctions, determiners, they include punctuation and words that play a role in a particular trim, like adverbs and gerunds. Other than that, the basic trimming task is done according to the following rules:

1. Remove extraneous words like date lines, editors comments etc.
2. Remove adverbs, conjunctions, preambles
3. Remove discourse particles like however
4. Remove ages mentioned in the article
5. Remove gerund phrases
6. Remove relative clause attributives
7. Remove mentions of quoted parties (police said), if the text is not a quote.

Developers erred on the side of rather missing one of these items than producing an ungrammatical sentence. The system produced less than 3% ungrammatical sentences, which is a great advance compared to the POS tagger trimmer used earlier, which had produced an ungrammatical sentence rate of 25% (Conroy, Schlesinger, O'Leary and Goldstein 2006: 3-4)

For document preparation, sentence splitting was also improved.

As for Query Term Selection, the extraction of relevant query terms was done by excluding words occurring in a function word list and a stop word list. Also, the query term algorithm was expanded by the use of stemming: This made it possible to include full words that correspond to stemmed versions of the words in the query term list. For the initial scoring and selection of sentences to be included in the summary, a high frequency measure for content words was carried out, as it is found in the SumBasic system. Instead of using the frequency of terms in order to estimate the likelihood of their appearance in human summaries, the new proposal is to directly model the set of terms that is likely to occur in a sample of human summaries. The variation that will be found in human summaries is modelled with a unigram bag-of-words model on the terms.

On the basis of a sample of human summaries, the developers compute the probability that a human will select a particular term in a summary with a given topic. For the estimation, query terms and signature terms are viewed as samples from idealized human summaries. Query terms are extracted from the topic description explained before, while signature terms are extracted from a set of sample documents. It is assumed that both query terms and signature terms will occur in a human summary. Signature terms are terms whose frequency in the summary document is higher than expected.

The summary is generated by using the top scoring sentences chosen from a set of sentences that contain at least eight distinct topical terms. In order to reduce redundancy, the summary produced from these top scoring sentences is first expanded to twice the target length, and then a pivoted-QR is used to select the final set of sentences. To determine the order of the sentences, a distance measure is applied in a Traveling Salesperson (TSP) formulation. Between each pair of sentences, a distance is defined on the basis of the number of terms they have in common. Then, an ordering is determined that minimises the sum of the distances between adjacent sentences. There is more than one possible choice for the distance function. Conroy, Schlesinger, O'Leary and Goldstein (2006) use similarity of terms between sentences, whereas Althaus, Karamanis, and Koller (2004) use a TSP formulation on the basis of a distance function that is a combination of coherence and salience. They also discuss other distance functions that have been used.

3.6. Document compression based on Centering Theory

Apart from being an ingredient in advanced abstract auto-summarization technology, sentence compression is recently treated as research field with significant potential for valuable impact. It can be used in a variety of applications besides auto summarization, including the generation of subtitles in television and on mobile devices, and generation of audio output of text for blind people (Nencova and McKeown 2011: 159; Clarke and Lapata 2007: 5; Clarke and Lapata 2010: 411; Cohn and Lapata 2009; Mc Donald 2006; Corston-Oliver 2001; Grefenstette 1998).

Integer Linear Programming (ILP), an approach used by Clarke and Lapata (2007), is generally statistical but applies discourse information as well, in order to create a model of sentence compression that is contextually aware. The linguistic theory used for the establishment of discourse information is Centering Theory (Grosz, Joshi, Weinstein 1994: 5-10). Centering Theory provides a model of the observation that for a sequence of utterances to form a discourse, the utterances must be coherent. The choice of referential expressions, the syntactic structure and the

word order are generally means to provide coherence in a discourse. Discourses can be coherent to different degrees.

A center is an entity that links the utterance to other utterances in the same discourse.

Each utterance has one backward-looking center Cb. This is the center that connects with one of the forward-looking centers of the following utterance. An utterance can have a number of forward-looking centers Cf. These are generally candidates for topichood, among which a ranked order is established in the utterance itself – contrary to the backward-looking center, they do not have any connection with the previous utterance. The backward-looking center is the topic of a sequence of utterances. Its definition entails the reappearance in an adjacent utterance as a prominent constituent, which is mostly a subject in pronominal form. If a sequence of at least two utterances provides a backward-looking center like this, the sequence has maximal discourse coherence.

Three types of transition relations between pairs of utterances in the discourse can be observed, reproduced here in a simplified explanation in the order of decreasing coherence (Grosz, Joshi, Weinstein 1994)[1]:

1. CENTER CONTINUATION

The backward-looking center of the current utterance is also the backward-looking center of the previous utterance, and the same entity is the most highly ranked center for topichood in the current utterance (CfMAX), which means that it is the most likely candidate to be the backward-looking center in the next utterance as well. This is a situation where a sequence of utterances is about the same entity, as in (8):

(8) John (Cb CfMAX) is the best student in his class. He (Cb CfMAX) has always been better than Kyle. He (Cb CfMAX) just enjoys school.

2. CENTER RETAINING

The backward-looking center is retained from one utterance to the next, but in the second utterance, it is no longer ranked as the most likely candidate for the following utterance.

(9) John (Cb, CfMAX) is the best student in his class. He (Cb) has always been better than this other guy, Kyle (CfMAX). Kyle (Cb) is rather lazy and likes playing football.

3. CENTER SHIFTING

The backward-looking center is not retained from one utterance to the next. There is a noticeable break in the coherence. A long discourse filled with utterance sequences like this would not be considered to be very readable.

(10) John (Cb) is the best student in his class. Kyle is rather lazy and likes playing

[1] Examples mine

football.[2]

Centering Theory makes a number of predictions about the ways in which backward-looking centers and the ranking of forward-looking centers can be established on the basis of the form of the referential expression chosen and the syntactic role that this referential expression occurs in. For our purposes, it is sufficient to conclude that Centering Theory is a potent dynamic description of topic coherence in discourse, and that it can be used for the implementation of discourse coherence in automatic summarization (Clarke and Lapata 2010: 423-426).

Clarke and Lapata use Centering Theory in their model for Sentence Compression with Discourse Constraints. They notice the paradox that summary applications always act on whole documents while sentence compression is carried out with isolated sentences (Clarke and Lapata 2010: 413).

In order to provide a summary with compressed sentences that represent the *flow of discourse* (Clarke and Lapata 2007: 7), a computation of local coherence is needed, and Centering Theory is chosen as the underlying linguistic theory, because it is an entity-oriented theory of local coherence. The coherence, according to this theory, is established in that coherent discourses have utterances with common centers. The authors also include the approach to lexical cohesion by Halliday and Hasan 1976, which establishes the topic on the basis of lexical chains. In order to arrive at a reliable computation of the center definition, the sentence is used as the unit corresponding to an utterance. For the C_f list, the authors use named entity recognition and coreference resolution, and for the detection of a C_b , they perform entity matching between sentences (Clarke and Lapata 2010: 425-427). The authors also use the grammatical roles of entities in order to establish the ranking. Subject is higher ranked than object, objects rank higher than other entities. Highest ranked entities of utterances are set to be the *center*, which is the backward-looking center and therefore the topic, in the following utterance. Using a lexical chain algorithm, semantically related expressions for topics are established.

For the actual compression, the statistical approach Integer Linear Programming (ILP) is used to incorporate constraints over the output in order to ensure that compressions are structurally and semantically valid. The constraints preserve grammaticality coherence by making sure certain words are retained, among which are: head words, centers, words in topical lexical chains and personal pronouns. Also, the sentential constraints involve retaining verbs and their core arguments, and making sure that each compressed sentence contains at least one verb. In the 2010 approach, the ILP is solved over the whole document instead for singular sentences.

The approach by Clarke and Lapata is a potent discourse-based sentence compression model, which is unsupervised and relatively simple in its application. In evaluation, the authors find that the performance of this system is better than the one of supervised discourse agnostic systems (Clarke and Lapata 2007: 29). As for the 2010 enhancement that uses a discourse ILP system, the

[2] There is a contradiction in the description of the backward-looking center and the explanation of the transition relations, in that the first word in each of these utterance sequences is called backward-looking center, even though by definition a backward-looking center is not selected within the utterance itself, but needs a predecessor. As these explanations only serve as a quick illustration of Centering Theory, it may be assumed that the entity *John* is established as a topic in the previous discourse and may therefore be called backward-looking center in these examples.

evaluation shows that coherence is improved, but the output summary is longer than the one produced by the sentence ILP (Clarke and Lapata 2010: 436-438).

4. Summary and outlook with some short remarks on summarization evaluation

The aim of this paper was to provide a short overview of some current approaches to automatic summarization. Many of the existing approaches are merely data-driven and unsupervised, and they do not require linguistic expertise or indeed any human intervention. While these extractive approaches are easy to compute and effective enough for many summarization tasks, the exploration of summarization methods does not stop with them. Abstractive summarization methods are more sophisticated in that they aim to come closer to a human-made summary. They involve sentence compression and a variety of methods to achieve topic continuity and coherence. In order to obtain these more advanced outputs, linguistic theories play an important role. Solutions that require syntactic information mostly rely on tree structures from Generative Grammar, while it is conceivable that functional theories of linguistics could be employed. Role and Reference Grammar (RRG) (Van Valin 2005), for example, provides a robust approach to argument structure, where the core verb and its main arguments are identified, but it does not separate syntactic and semantic information. Rather, the argument structure is built as a consequence of basic semantic information found in the verb, which is its *Aktionsart*. Furthermore, RRG makes it very clear which parts of its syntactic specifications and dependencies are language specific, and which are valid cross-linguistically. RRG can therefore be applied to different languages. This functional theory of linguistics has already been used for syntactic parsing, Machine Translation and other NLP tasks (Nolan and Periñan-Pascual 2014) and for a concept of Conversational Agents (Nolan, this volume).

As for the implementation of topic continuity and coherence, Centering Theory has been found to be very useful in building a summarization system that is context aware and provides a coherent and readable output text. Automatic summarization systems are subject to evaluation, and there is a wealth of research and literature on the topic of summarization evaluation itself. Summarization evaluation is not a trivial task, and there cannot be a one-fits-all solution, either. Given the large variety of textual input that is undergoing summarization, and the different requirements the summaries are to fulfil, there is a large number of approaches to summarization evaluation (Spärck Jones 2007; Mani 2001; Nenkova and McKeown 2011; Jurafsky and Martin 2009). Automatic Summarization, along with its evaluation, has come a long way since its first applications in the 1950s, but it is still a growing field. Its applications and challenges are expanding with the growing number of textual information produced and shared via the Internet, and with the demands posed by mobile and wearable devices, and other voice-activated appliances with access to massive data sources from the cloud. New strategies for automatic summarization will also be critical for information retrieval from the semantic web, and it will require linguistic processing capabilities (Diedrichsen 2016; Nolan 2016) in as many languages as possible.

References

Althaus, Ernst, Nikiforos Karamanis and Alexander Koller. 2004. Computing locally coherent discourses. *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, 399-406. https://doi.org/10.3115/1218955.1219006

Baum, Leonard E. 1972. An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process. *Inequalities*, 3: 1-8.

Baum, Leonard E., Ted Petrie, George Soules, and Norman Weiss. 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics* 41: 1, 164-171. https://doi.org/10.1214/aoms/1177697196

Chin-Yew Lin and Eduard Hovy. 2000. The automated acquisition of topic signatures for text summarization. *Proceedings of the International Conference on Computational Linguistics*, 495-501. https://doi.org/10.3115/990820.990892

Clarke, James and Mirella Lapata 2010. Discourse Constraints for Document Compression. *Computational Linguistics* 36: 3, 411-441. https://doi.org/10.1162/coli_a_00004

Clarke, James and Mirella Lapata. 2007. Modelling Compression with Discourse Constraints. *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 1-11.

Cohn, Trevor and Mirella Lapata. 2009. Sentence compression as tree transduction. *Journal of Artificial Intelligence Research*, 34: 637–674.

Conroy, John M., Judith D. Schlesinger and Jade Goldstein Stewart. 2005. CLASSY query-based multi-document summarization. *DUC 05 Conference Proceedings.*

Conroy, John M., Judith D. Schlesinger, Dianne P. O'Leary and Jade Goldstein. 2006. Back to Basics: CLASSY 2006. *Proceedings of the Document Understanding Conference*, 2006.

Corston-Oliver, Simon. 2001. Text compaction for display on very small screens. *Proceedings of the NAACL Workshop on Automatic Summarization*, 89–98.

Diedrichsen, Elke (2016): Does NLP need Theoretical Linguistics? In Periñan-Pascual, Carlos and Eva M. Mestre-Mestre (eds.): *Understanding Meaning and Knowledge Representation: From Theoretical and Cognitive Linguistics to Natural Language Processing*. Newcastle Upon Tyne: Cambridge Scholars Publishing, 249-258.

Diedrichsen, Elke. 2014. A Role and Reference Grammar Parser for German. In Nolan, Brian and Carlos Periñan-Pascual (eds.): *Language Processing and Grammars. The Role of Functionally Oriented Computational Models.* Amsterdam: John Benjamins, 105-142. https://doi.org/10.1075/slcs.150.05die

Filatova, Elena and Vasileios Hatzivassiloglou. 2004. A formal model for information selection in multi-sentence text extraction. *Proceedings of the International Conference on Computational Linguistics*, 397-403. https://doi.org/10.3115/1220355.1220412

Grefenstette, Gregory. 1998. Producing Intelligent Telegraphic Text Reduction to Provide an Audio Scanning Service for the Blind. *Proceedings of the AAAI Symposium on Intelligent Text Summarization*, 111–117.

Grosz, Barbara J., Aravind K. Joshi and Scott Weinstein. 1994. Centering: a framework for modeling the local coherence of discourse. *University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS: 94-40*, 1-27.

Halliday, M. A. K. and Ruqaiya Hasan. 1976. *Cohesion in English*. Longman, London.

Hatzivassiloglou, Vasileios, Judith L. Klavans, Melissa L. Holcombe, Regina Barzilay, Min-Yen Kan and Kathleen R. McKeown. 2001. SIMFINDER: A flexible clustering tool for summarization. *Proceedings of the NAACL Workshop on Automatic Summarization*, 41-49.

Jing, Hongyan and Kathleen R. McKeown. 2000. Cut and paste based text summarization. *Proceedings of the North American chapter of the Association for Computational Linguistics Conference*, 178-185.

Jing, Hongyan. 2000. Sentence reduction for automatic text summarization. *Proceedings of the Conference on Applied Natural Language Processing*, 310-315. https://doi.org/10.3115/974147.974190

Jurafsky, Daniel and James H. Martin. 2009. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* 2nd edition. Pearson Education Inc.

Lewis, David Dolan. 1999. An evaluation of phrasal and clustered representations on a text categorization task. *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR* 1992), 37-50.

Luhn, H. P. 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2: 2, 159-165. https://doi.org/10.1147/rd.22.0159

Mani, Inderjeet. 2001. Summarization Evaluation: An Overview. *Proceedings of the NTCIR Workshop 2 Meeting on Evaluation of Chinese and Japanese Text Retrieval and Text Summarization*. Tokyo: National Institute of Informatics.

McDonald, Ryan. 2006. Discriminative sentence compression with soft syntactic constraints. *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, 297–304.

Morris, Jane and Graeme Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics,* 17(1): 21–48.

Nolan, Brian (2016): What can Theoretical Linguistics do for Natural Language Processing Research? In Periñan-Pascual, Carlos and Eva M. Mestre-Mestre (eds.): *Understanding Meaning and Knowledge Representation: From Theoretical and Cognitive Linguistics to Natural Language Processing*. Newcastle Upon Tyne: Cambridge Scholars Publishing, 235-248.

Quazvinian, Vahed and Dragomir R. Radev. 2008. Scientific paper summarization using citation summary networks. *Proceedings of the International Conference on Computational Linguistics,* 689-696. https://doi.org/10.3115/1599081.1599168

Rabiner, Lawrence E. 1989. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE,* 77: 2, 257-286. https://doi.org/10.1109/5.18626

Schwartz, Richard, Toru Imai, Francis Kubala, Long Nguyen and John Makhoul. 1997. A maximum likelihood model for topic classification of broadcast news. *Proceedings of the Fifth European Speech Communication Association Conference on Speech Communication and Technology (Eurospeech-97).*

Siddarthan, Advaith, Ani Nenkova, and Kathleen R. McKeown. 2004. Syntactic simplification for improving content selection in multi-document summarization. *Proceedings of the International Conference on Computational Linguistics,* 896-902. https://doi.org/10.3115/1220355.1220484

Sista, Sreenivasa, Schwartz, Richard, Leek, Timothy R. and John Makhoul. 2002. An algorithm for unsupervised topic discovery from broadcast news stories. *Proceedings of the 2002 Human Language Technology Conference (HLT),* 99-103. https://doi.org/10.3115/1289189.1289267

Spärck Jones, Karen. 2007. Automatic summarising: The state of the art. In *Information Processing and Management.* 43 (2007) 1449–1481. https://doi.org/10.1016/j.ipm.2007.03.009

Van Valin, Robert D. Jr. 2005. *Exploring the Syntax-Semantics Interface.* Cambridge: Cambridge University Press. https://doi.org/10.1017/CBO9780511610578

Yih, Wen Tau, Joshua Goodman, Lucy Vanderwende and Hisami Suzuki. 2007. Multi-document summarization by maximizing informative content-words. *Proceedings of the International Joint Conference on Artificial Intelligence,* 1776-1782.

Zajic, David, Bonnie J. Dorr, and Richard Schwartz. 2004. BNN/UMD at DUC-2004: Topiary. *Proceedings of the 2004 Document Understanding Conference (DUC 2004)* at NLT/NAACL 2004, 112-119.

Zajic, David, Bonnie J. Dorr, Jimmy Lin and Richard Schwartz. 2007. Multi-candidate reduction: Sentence compression as a tool for document summarization tasks. *Information Processing and Management*, *43:6*, 1549-1570. https://doi.org/10.1016/j.ipm.2007.01.016