From Extractive to Abstractive Summarization: A Journey

by

PARTH MEHTA
201321005

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY
to

DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY

May, 2018

Declaration

I hereby declare that

i) the thesis comprises of my original work towards the degree of Doctor of Philosophy at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,

ii) due acknowledgment has been made in the text to all the reference material used.

Parth Mehta

Certificate

This is to certify that the thesis work entitled FROM EXTRACTIVE TO ABSTRACTIVE SUMMARIZATION: A JOURNEY has been carried out by PARTH MEHTA for the degree of Doctor of Philosophy at *Dhirubhai Ambani Institute of Information and Communication Technology* under my supervision.

Prasenjit Majumder
Thesis Supervisor

Acknowledgments

First of all I would like to thank Prof. Prasenjit Majumder for his valuable guidance and encouragement throughout my PhD. Without him this it would have taken me a few more years to complete this journey. An equal thanks to my parents and my siblings for their never ending patience, belief and support, and for never questioning my choice of pursuing PhD. I would also like to thank the members of my research progress seminar committee, Prof. Jaideep Mulherkar and Prof. Bharani Kollipara; my thesis reviewers Prof. Jaap Kamps and Dr. L V Subramanian; and members of my PhD defense committee, Prof. Suman Mitra and Prof. Pankaj Kumar, for their valuable feedback throughout my PhD journey.

I am grateful to all the members of IRLAB, present and past, for their support and help. You have been a great team and the journey wouldn't have been so much fun without you. I would also like to thank all the faculty members of DAIICT and my fellow Phd students for their love and encouragement through all these years. A special thanks to my room mates Nirmesh Shah, Sumukh Bansal, Kamal Captain and Gaurav Arora, for tolerating all my antics and for always being there for me.

Contents

Abstract

Research in the field of text summarisation has primarily been dominated by investigations of various sentence extraction techniques with a significant focus towards news articles. In this thesis, we intend to look beyond generic sentence extraction and instead focus on domain-specific summarisation, methods for creating ensembles of multiple extractive summarisation techniques and using sentence compression as the first step towards abstractive summarisation.

We start by proposing two new datasets for domain-specific summarisation. The first corpus is a collection of court judgements with corresponding handwritten summaries, while the second one is a collection of scientific articles from ACL anthology. The legal summaries are recall-oriented and semi-extractive, compared to the abstracts of ACL articles which are more precision oriented and abstractive. Both collections have a reasonable number of article-summary pairs, enabling us to use data-driven techniques. Excluding newswire corpora where the summaries are usually article headlines, the proposed collections are amongst the largest openly available collections of document summarisation. Next, we propose a completely data-driven technique for sentence extraction from legal and scientific articles. In both legal and ACL corpus, the summaries have a predefined format. Hence, it is possible to identify *summary worthy* sentences depending on whether they contain certain key phrases. Our proposed approach based on attention-based neural network learns to automatically identify these key phrases from pseudo-labelled data, without requiring any annotation or handcrafted rules. The proposed model outperforms existing baselines and state of the art systems by a large margin.

There are a large number of sentence extraction techniques, none of which guarantee better performance than the others. As a part of this thesis, we explore if it is possible to leverage this variance in performance for generating an ensemble of several extractive

techniques. In the first model, we study the effect of using multiple sentence similarity scores, ranking algorithms and text representation techniques. We demonstrate that such variations can be used for improving Rank Aggregation. Using several sentence similarity metrics, with any given ranking algorithm, always generates better abstracts. Next, we propose several content-based aggregation models. Given the variation in performance of extractive techniques across documents, the apriori knowledge about which technique would give the best result for a given document will drastically improve the result. In such case, an oracle ensemble system can be made which chose best possible summary for a given document. In the proposed content-based aggregation models, we estimate the probability of a summary being good by looking at the amount of content it shares with other candidate summaries. We present a hypothesis that a good summary will necessarily share more information with another good summary, but not with a bad summary. We build upon this argument to construct several content-based aggregation techniques, achieving a substantial improvement in the Rouge scores.

In the end, we propose another attention based neural model for sentence compression. We use a novel context encoder, which helps the network to handle rare but informative terms better. We compare the proposed approach to some sentence compression and abstractive techniques that have been proposed in past few years. We present our arguments for and against these techniques and build a further roadmap for abstractive summarisation. In the end, we present the results on an end to end system which performs sentence extraction using standalone summarisation systems as well as their ensembles and then uses the sentence compression technique for generating the final abstractive summary.

List of Tables

List of Figures

CHAPTER 1

Introduction

Automatic Summarization, or reducing a text document while retaining its most essential points, is not a new research area. The first notable attempt, which dates back to 1958, was made by [44]. It uses word frequencies to identify significant words in a given sentence. The importance of a sentence is then determined from the number of significant words it has and proximity of these words to each other. Since then the techniques for both sentence selection (extractive summarisation) as well as abstract generation (abstractive summarisation) have advanced a lot. Unfortunately, most of the initial works were either not reproducible due to lack of standard evaluation corpora [5], [38] or worse, they were not evaluated at all. It was only after the advent of conferences like DUC[14] and TAC[60], which generated standard evaluation benchmarks for text summarisation, that streamlined efforts were made possible. In this chapter, we begin by providing a brief overview of the types of summarisation systems and their positive and negative aspects. Next, we highlight the main contributions of this thesis, and in the last section, we provide a brief overview of rest of this thesis.

The survey by [15] broadly categorises these techniques into Extractive, Abstractive and Information Fusion techniques. Extractive techniques, as the name suggests, solely rely on extracting essential sentences or phrases from a document or set of documents. The Abstractive techniques focus on rewriting the content in a more precise manner. Most state of the art abstractive systems focus on paraphrasing or sentence simplification to achieve this, and though there have been some attempts at sentence generation, they have been of limited success to date. Sentence compression techniques have evolved as a bridge between sentence extraction and abstractive techniques. These are still closer to the extractive techniques, but also perform operations like phrase deletion[35] to shorten the

extracted sentences and sentence fusion to combine two or more related sentences [4].

1.1 Extractive summarization

Extractive summarisation is perhaps the most researched category amongst the three types of summarisation systems. The term is generally used for sentence extraction and reordering, but several extractive techniques also focus on sub-sentence level extraction. There are three main variants of an extractive system topic base and centrality based. The topic-based systems focus on assigning relative importance to individual words or phrases. A sentence is then ranked based on the aggregate importance. The first type of systems can define the weights in different ways, like simple frequency based weights[44], or by identifying important terms by comparing the document with a background corpus like in [41] or by maintaining a dictionary to give more importance to terms related to a particular domain. In contrast, centrality based techniques try to identify the most important sentence by ranking them based on the content they share with other sentences. Several popular techniques like Centroid-based technique[67], LexRank[20] and TextRank[50] are centrality based. The key idea is to find sentences that best represent the overall information in a document. The third category, which has proved to be quite successful, focuses on the overall target summary rather than individual sentences. These methods look at summarisation as an optimisation problem for selecting the best possible subset from a collection of sentences[29]. All these methods solely aim at improving the coverage of summaries. The importance of a sentence solely depends on the information it adds to the summary.

This is not always the ideal case when it comes to summarising documents from a specific domain. For example, in case of summarising multiple news articles, *newness* of the information might be more important than coverage. A reader following a particular event will already be up to date with the major highlights and would be more interested only in the *'breaking news'* related to the event. Several factors affect such extractive systems many of which, like the background knowledge of the user, might not always be easy to capture, if at all possible. However, in some instances, the summaries follow a predefined template, and it is known beforehand that only *certain* type of information will be useful in the summary. Most existing techniques for template-based summarisation rely on human

inputs for creating the templates and then later select sentences based on whether or not they fit into the template[61]. Creating such templates requires a lot of human efforts. In this thesis, we demonstrate how a simple LSTM based encoder, with an attention module, can efficiently learn to create such extractive summaries. We present two domain-specific corpora, related to legal documents and scientific articles. In both cases, the nature of summaries is such that they implicitly follow a template. For instance, in scientific articles, it is almost always the case that an abstract starts with defining the problem, followed by a proposed solution, some key results and usually has some information about the corpus used. Similarly, in case of legal documents, certain phrases that indicate essential information are critical factors in a sentence being *summary worthy*. Besides this, defining the overall context of the article is essential for sentence selection. We show that the proposed model is very efficient at identifying the key phrases which play a pivotal role in sentence selection. This is similar to template matching based summarisation, except in this case the template is not pre-defined but learnt from the available data.

1.2 Information fusion and ensemble techniques

Ever-Increasing interest in the field of automatic summarisation has led to a plethora of extractive techniques. Unfortunately, this is often accompanied by lack of clarity about how good each system is compared to the rest. Several studies highlight the variance in performance of these systems with a change in datasets or even across documents within the same corpus. This difference is often attributed to unclear implementation details and variation in evaluation setups, which can result in substantial variation in the scores. The work by [31] shows that even with normalisation of results by using a standard evaluation setup and a fixed set of parameters for ROUGE, the system performance still varies a lot, and there is a possibility of exploiting this variation to generate better-performing systems. There is no doubt that none of these techniques would always outperform the others. An effective way to counter this variance and to make the systems more robust could be to use inputs from multiple systems when generating a summary.

In the present thesis, we define two novel methods of creating such ensembles. In the first method, we focus on highlighting the effect of variation in the three principal

components of any sentence extraction technique: sentence ranking algorithm, sentence similarity metric and text representation scheme. We show that such variations can be exploited to improve the overall coverage of extractive summaries. Our experiments show a significant improvement in terms of ROUGE score when using several sentence similarity metrics. Such ensembles are also much more robust compared to the original systems. However, these ensembles are still not better than the Oracle systems, where we select the best performing candidate system for each document. In the second approach, we look at estimating the reliability of individual systems and use that to weight content from each system. We use the candidate summaries as pseudo-relevant summaries. We propose a consensus-based aggregation technique, which takes into account the content the content of candidate summaries and uses the overlap between them to estimate their reliability. We define GlobalRank which captures the performance of a candidate system on an overall corpus and LocalRank which estimates its performance on a given document cluster. We then use these two scores to assign a weight to each system, which is used to generate the new aggregate ranking. Experiments on DUC2003 and DUC 2004, as well as the Legal and ACL datasets, show a significant improvement in terms of ROUGE score, over existing state-of-art techniques.

1.3 Abstractive summarization

As compared to the extractive techniques, abstract generation approaches seem more natural and closely relate to the way humans summarise the documents. Numerous attempts have been made at creating abstractive summaries, most of which have been limited to headline generation from news articles. The work by [3] is somewhat similar to our proposed machine translation based sentence compression model. A significant difference is that our model works on much longer sentences compared to average 4-5 word news headlines in [3]. Many of these approaches rely on linguistic resources and use the syntax of the original and compressed sentences. The work by [11] learn sentence transformations using tree transduction rules on aligned and parsed sentences. Similarly, the work by [81] uses quasi-synchronous grammar which depends on context-free parsing as well as dependency parsing. However, the success in abstract generation has mostly been limited due to

the inability of present computational techniques to generated very fluent and grammatically correct abstractive summaries, as well as their dependency on linguistic resources. Given these limitations, most practically viable abstractive summarisation system use sentence extraction techniques to form an initial extractive summary, and then use sentence compression, simplification or sentence fusion to form the final abstract. Lately, as more training data becomes available and with the recent advances in the field of deep learning, some fresh attempts like [68], [9] and [8] are now being made towards data-driven abstractive summarisation that has minimal dependency on linguistic inputs.

In this thesis, we present two approaches based on machine translation techniques, which can be used for sentence compression under specific constraints. We show that for domain-specific data, like the legal documents, where the vocabulary is limited, it is possible to use phrase-based translation models to generate sentence compressions. Next, we use the attention based sequence to sequence model for generating sentence compressions. This is inspired by the neural machine translation model. The sequence model based approach inherently uses the global context of sentence form last based encoder, while the attention module provides the local context. These are together used to decide which phrase needs to be deleted or simplified. We compare both these approaches to the deletion based compression approach of [21]. Compared to the deletion based approach, both our methods perform well and achieve higher accuracy in the generated sentence compressions.

1.4 Main contributions

In this section, we provided a brief overview of the main contributions of this thesis. While some problems addressed in this thesis have been actively researched upon for a long time, some of them have received little attention. With increasing amount of data becoming available and the computing capability increasing many-fold in the past few years, the focus is now more towards data-driven techniques. As such this thesis focuses on techniques with minimal human or linguistic inputs. Following are the primary contributions:

- We propose two new corpora for domain-specific summarisation. The datasets contain Legal and Scientific articles, which are amongst the most extensive publicly

available corpora. The legal documents are such that the summaries are semi-extractive, which makes them suitable for investigations in both extractive and abstractive techniques.

- Next we propose an attention-based sentence extraction technique which is capable of automatically identifying cue phrases that make a sentence summary worthy. Our technique was able to outperform even those extractive techniques which incorporated domain knowledge.

- After domain specific sentence extraction we propose another method to improve the performance of existing extractive techniques. We point out that the performance of existing techniques vary across documents, and none of the methods is always better than the other. Instead, we propose using inputs from multiple systems and generate an aggregate ranking. The proposed ensemble technique can leverage multiple sentence similarity scores, sentence ranking algorithms and text representations schemes to improve the coverage as well as the robustness of aggregate summary.

- We complement the Rank aggregation based ensemble proposed above, with content-based aggregation techniques which can estimate the reliability of a candidate system for a given document. We propose multiple techniques to estimate which of the given candidate systems will perform better for a given document. This knowledge can then be used to give relative weights to the systems when aggregating their outputs.

- At the end, we propose a sequence to sequence model based sentence compression technique that can shorten and simplify sentences. We contrast this to various approaches and provide a further roadmap for sentence compression and abstract generation techniques.

1.5 Thesis organization

The rest of this thesis is organised as below. We discuss the existing approaches for domain-specific summarisation, neural summarisation and ensemble techniques as well as some of their limitations in the second chapter. In chapter 3, we discuss in detail two

particular cases of domain-specific summarisation, i.e. legal document summarisation and scientific articles summarisation. We highlight how the nature of summaries varies across these two domains and the shortcomings of existing summarisation techniques in handling these variations. We also present the two datasets that we created as a part of this thesis. We then explain the proposed weakly supervised neural model for sentence extraction. We define a new method for incorporating the document context in a neural summarisation setup using topic models. We contrast this more straightforward approach for document level encoding to the commonly used methods that use LSTMs to encode the entire documents. We show that not only is our approach much faster, it also less reliant on the amount of training data. We also show how a simple word level attention model can detect essential cue phrases automatically, without the use of any dictionary or explicit tagging.

In the fourth and fifth chapters, we discuss two approaches for creating an ensemble from several different summarisation systems. We begin by pointing out that no particular sentence extraction technique is better than the other in all the cases. We argue that a natural solution to this would be using inputs from multiple systems to generate an aggregate summary. We propose two approaches to achieve this. In the fourth chapter, we highlight how various components of a summarisation system can affect its performance. We further show that variations and combinations of these fundamental components can be used to create several candidate systems, which when combined will produce a summary with much better coverage. We proposed using several sentence similarity metrics, ranking algorithms and text representation schemes to generate multiple sentence rankings. Such rankings can then be combined using rank aggregation techniques to produce better summaries. We compare the proposed methods on the standard DUC datasets as well as on the new legal and scientific corpora. Next, we discuss the limitations of rank aggregation techniques in general and emphasise on several issues which they face. Next, we propose several content-based aggregation approaches, which take into account the content of candidate summaries rather than merely looking at ranked lists. We present a hypothesis that, in the absence of ground truth summaries, it is possible to use several candidate summaries as pseudo-relevant summaries. We define several measures to determine this reliability of a system at a global as well as at local level. Finally, we combine these measures into a

hybrid ranking which is used to estimate the overall reliability of a system for a given document. As in previous chapters, we compare the results on the legal and scientific corpora, apart from DUC datasets.

In the sixth chapter, we propose two sentence compression models that are inspired by machine translation models in general. The sentence extraction models introduced in the previous sections are complemented with these sentence compression models to generate abstractive summaries. First, we propose a modification of the phrase-based translation model, which is capable of translating a *document sentence*, which is longer, to a *summary sentence* which is relatively compact. Next, we describe the neural compression models which are capable of achieving the same goal. We discuss the pros and cons of each method and compare them by evaluating them on the corpora mentioned in previous chapters. We also compare this to the deletion based sentence compression method.

Finally, we conclude the thesis in the last chapter where we provide an insight into the possible shortcomings as well as possible extensions of this thesis.

CHAPTER 2

Related work

In this chapter, we examine some of the existing techniques for sentence extraction and sentence compression. We also discuss the existing approaches for creating ensembles of these systems. We point out specific cases where the existing techniques would not work and what needs to be done to handle such cases.

2.1 Extractive summarization

Extractive summarisation is a widely researched area, and several techniques are worth mentioning. However, in this thesis, we describe such techniques which maximise the representation from several different categories of extractive techniques. A survey by [57] categorises the extractive summarisation system based on whether they are topic representation based or indicator representation based. A topic representation based system can vary from as simple representation as tf-idf [67] or word frequency [58], to topic signatures[41] or latent semantic indexing. Other alternative text representations are term distributions [29] or Latent semantic indexing [75]. Some recent attempts, like [36] and [34], have used word embeddings based representation. As opposed to these, indicator representation approaches do not rely on extracting or interpreting topics. They instead represent a document in a way that direct ranking of sentences becomes possible. Some well-known approaches of this kind are the popular graph based approaches[20, 50].

Another popular classification of the summarisation systems is based on their ranking algorithms. We choose the three most popular algorithms: centrality based, corpus-based and graph based. Centrality based techniques usually rely on a *abstract* sentence that is central to the given document or set of documents. The idea is then to iteratively find

sentences that are most similar to this *central* sentence and include them in summary. Such a system does not take into account the summary produced so far and hence is susceptible to redundancy. Often the highly ranked sentences are quite similar to each other, and a separate mechanism to handle redundancy is required.

In the Graph-based techniques, document (or set of documents) are represented as a sentence graph. Each sentence constitutes a node, and the edges represent similarity between sentences. There are many ways to leverage this representation for generating summaries. One popular method of achieving this is by looking at the number of nodes connected to a given node. [69] define bushiness of a node as the number of links connecting it to other nodes. It also takes into account natural cohesiveness of text segments to generate a coherent summary. A global bushy path is created using the N most bushy nodes, and then these nodes are arranged in chronological order to obtain the summary. Lexrank[20] is another immensely popular graph-based technique. In this method, the nodes in a document graph are weighted depending on their *reputation*, which in turn depends on the number of nodes it is connected to and the reputation of those nodes. The reputation of each node is computed iteratively. In reality, Lexrank is an overlap between graph based and centrality based techniques. Inherently Lexrank computes lexical centrality of sentences and selects the sentences which have maximum common words with other sentences.

In contrast to graph-based and centrality based techniques, which do not rely on any other information except the documents themselves, some techniques make use of a background corpus while ranking the sentences. One such technique proposed by [58] estimates of word frequencies using a background corpus, and then ranks sentences based on the number of highly frequent words that it contains. The summarisation technique proposed by [41] uses a background corpus to acquire *Topic Signatures*, which are nothing but the words used more frequently in a given document compared to a background corpus. The sentences having more topic signatures are ranked higher. Both these approaches do not take into account the similarity between sentences.

Extractive techniques like LexRank[20] and TextRank[50] try to select the best representative set of sentences from a given document/document cluster without any modification. In such techniques, each document is represented as a graph, and each sentence

in the document constitutes a node. The sentences are ranked using an approach similar to the popular Pagerank algorithm [62]. However, nonetheless, even indicator representation based techniques will require a text representation scheme when computing sentence similarity scores.

All the approaches mentioned so far only consider the document or an additional background corpus, and iteratively select sentences to be included in the summary. None of them takes into account the summary generated so far for selecting the subsequent sentences. In contrast, the greedy approach iteratively selects sentences to minimise the divergence between summary term distribution and document term distribution at each step. GreedyKL proposed in [29] uses KL divergence to compute the similarity between the summary and document.

We select representative techniques from each of these categories. For our experiments, we use five candidates systems Lexrank and TextRank which are both graphs based and indicator representation based technique. Centroid[67], TopicSum[13] and Greedy-Kl[29] are all topic representation based techniques. While centroid is centrality based, topicSum is a topic based system. We also compare the results with some state of the art techniques like Detrimental point processing[37], Integer Linear Programming[10] and Submodular[43], which we describe in later sections.

All these extractive techniques solely focus on identifying the important content in a document, without taking into account the end goal. For example, as we show later, there are cases where coverage of a summary is not essential, but the focus is on specific aspects of the documents. In such cases, these techniques would not work, and usually, a pre-defined template is required[61]. As a part of this thesis, we propose a data-driven model to generate such templates automatically, thus eliminating the need for any manual annotation.

2.1.1 Legal document summarization

Most previous works towards legal document summarisation focus on identification of rhetorical roles and then using this information to generate summaries [26, 71]. The work by [26] provides a general framework for identifying rhetorical roles in legal judgements. They define seven rhetorical roles: act, proceedings, background, proximation, distancing,

framing and disposal. Each sentence can then be categorised into one of these roles using a set of rules along with POS, chunking and tense identification. The work by [72, 74] define a different set of rhetorical roles from [26]. Neither of them provides any justification as to why these particular roles. [72, 74] then use CRF for labelling each sentence with a particular rhetorical role. They separately use a K-mixture model to identify important sentences in the judgement. Next, they use handcrafted rules for giving more weights to the sentences with particular rhetorical roles. The first work [72] reports only f-score for labelling problem, while the summaries are not evaluated at all. The subsequent follow up works in [74, 71] use ROUGE-1 and ROUGE-2 for evaluating the summaries. Due to lack of implementation details, we do not report the results on this technique. The work by [73] focuses on the building of legal ontology to assist summarisation. The ontology focuses on six main categories: group, person, things, event, facts and acts. The authors then define several relations like is-a, related-to, composed-of, etc. to form relations between different terms. [54] argue on similar lines as us. They believe cue-phrases and prior knowledge of text structure would play a crucial role in automatically generating abstracts. They use cue phrases as a signal for boundaries of various logical sections in a judgement. They then segment the documents in various classes like *accused*, *victim*, *alleged_offesne*, *opinion_of_the_court*, etc. In the proposed approach we do not identify these sections explicitly, but the attention module inherently learns to identify such segments and use that information to rate a sentence. Another work relevant to the proposed approach is by [45]. They propose a rule-based method for identification of Legal catchphrases. This is a somewhat different approach from rhetorical role labelling, in a sense, this does not limit the possible catchphrases to a fixed number of classes. Instead, the technique focuses on determining how relevant a phrase is to a given document. We incorporate this method into a sentence extraction setup and use it as a baseline for comparison.

Overall, all these represent diverse approaches to solving the same problem, which is identifying essential phrases in a legal document. Many rules for identifying the rhetorical roles depend on such keyphrase. Even the legal ontology framework depends on the same. In this thesis, we propose an attention-based neural model that can automatically identify such phrases which are implicitly used to decide if a sentence is *summary worthy*.

2.1.2 Scientific article summarization

A major focus of scientific article summarization has been towards using citation information for identifying relevant content. The focus in such cases, is to use the information from target articles to identify contribution of a given paper[65, 66]. These works define citation summary of article *A* as the set of sentences which cite the article. A citation summary network is then defined as a network in which each sentence of the citation summary is a node and the edge represents similarity between the two nodes. They then use graph clustering methods to cluster the citation summary network. Various techniques like cluster-lexrank and round robin are used to identify important sentence from each clusters. The idea in such cases is to highlight the important contributions of the article in a summary. This is slightly different from general abstract of an article, which will have some information about the problem statement, datasets used and evaluation strategy. The work by [49] uses a similar approach. They define a *Impact Language Model*, to reflect sentences which are more probable to be impactful and cited by other articles. They present an argument that it is possible to represent a summary worthy sentence by a specific language model. This is in fact generalization of our hypothesis that such sentences can be identified using cue-phrases. The approach is interesting, but it relies on handcrafted gold standard summaries, which can vary drastically depending on the human annotator. At the same time, their dataset contains only 14 such summaries, which somewhat limits its usability. [1] builds upon previous works related to citation networks. Unlike the techniques in [65, 66] which solely focus on the *informativeness* os individual sentences, this technique also considers readability, diversity and coherence of the final summary. The work by [78] assumes a more traditional approach, which was very popular in legal document summarization, that of identifying rhetorical roles and relevance of sentences. The define several roles like *Aim*, *section information*, *backgorund*, etc. Each sentence in an article is then labelled with one of these roles and also with a separate binary label, indicating whether or not it is useful for inclusion in summary. Besides this, they use several other features like the position of the sentence, sentence length, title words, etc. They also identify several indicators which are similar to what we define as the cue phrases, e.g. *'in this paper'*, *'when compared to our'*, etc. They use all this information to train a classifier. The approach proposed by us in the current work tries to attempt the same, by automatically

learning such important features. We show later in the result that our system automatically gives importance to several such cue phrases when deciding whether or not a sentence is a summary worthy. The work in [30] focuses on identifying sections in an abstract using conditional random fields. They identify rhetorical roles of sentences within an abstract and classify them into *objective*, *method*, *results* and *conclusion*. Overall the work presented in this thesis is different from all these previous attempts in two ways: one it tries to generate an actual abstract and not just impact based summary, and two it automatically learns the cue-phrases, while almost all the other methods rely on manual tagging for this information.

2.2 Ensemble techniques for extractive summarization

In contrast to the amount of attention extractive summarisation has received, aggregation techniques have been explored little. Few attempts have been made at combining various existing approaches. The existing approaches can be broadly classified into two categories. Pre-summarization ensemble techniques incorporate features or ranking techniques from several approaches to directly generate an aggregate summary. As opposed to this, the post-summarisation techniques look at aggregating the summaries or ranked list of sentences generated by candidate systems.

We could find only three previous works that are directly relevant to the proposed approach. Two of these approaches, [80] and [64], assume the availability of candidate summarisation systems and hence that of the original rank list of sentences generated from these candidate techniques. Most extractive summarisation systems first rank the sentences according to their importance and then later reorder them as per their original order in the document. In this case, it is not always possible to generate partial rank lists of sentences only from the original document and the summary without access to the actual summarisation system. As opposed to that, the most recent approach[32] only requires the summaries and the original documents, but not the original rank lists. This works well when the goal is to report results on a commonly used benchmark dataset. However, in practice, such approach would still require the original systems, even if as a black box, to generate the summaries. As an alternative, we propose creating variations of a single

system, by using several sentence similarity metrics or ranking algorithms, and then ensemble them to create a meta-system that has higher overall efficiency. A similar system is proposed by [70] where they demonstrate the effect of using more than one sentence similarity score for query expansion. They show that for query expansion, using an ensemble of a wordnet based semantic score and the vector space based similarity metrics outperform the individual systems by a large margin.

The first attempt of creating an aggregate summary was made by[80] where they take a weighted combination of rank lists from four candidate systems and produce a new rank list which can then be used to produce an extractive summary. They treat this as an optimization problem trying to reduce the weighted distance between candidate summaries and resultant aggregate, under a constraint of smoothness of weights. Given ranklists $r_1, ..., r_N$ the paper tries to find out optimal weights $w_1, ..., wN$ such that the new ranklist (r^*) is as close as possible to each of the original ranklists, while imposing a constraint of smoothness on the weights w_i. This translates into iteratively solving the following equations:

$$\textbf{Step 1: } r^* = \sum_{i=1}^{n} (w_i r_i) \tag{2.1}$$

$$\textbf{Step 2: } w_{optimal} = \underset{w}{\operatorname{argmin}} (1-\lambda) \sum_{i=1}^{N} w_i ||r^* - r_i||^2 + \lambda ||\mathbf{w}||^2 \tag{2.2}$$

We use this technique as one of the aggregation methods in our experiment. The paper by [80] focuses on finding out a better aggregation technique compared to the existing ones like Borda count or correlation based weighting. In contrast to finding alternates for Borda or weighted consensus methods, this thesis focuses on highlighting the conditions under which performance of an existing aggregation system can be improved. In this sense, the proposed approach complements weighted consensus summarisation and other existing aggregation techniques. As an example, we show that all these aggregation techniques perform much better when rank lists generated by the same system but using different similarity scores are used, as compared to combining rank lists from very different systems.

Another approach demonstrated in [64] uses SVM-Rank for generating the aggregate summaries. For generating the required labelled data, they use the following approach: for every n-gram t_n (n = 1,2) in H (human-made summary) $P(t_n|H)$ is computed, and each sentence is scored as the sum of these probabilities for all n-grams. Sentences above a threshold are considered to be essential and included in the summary. This data is then used to train an SVM-Rank system to select the best ordering of sentences present in the candidate summaries. One major problem with this approach is the availability of training data. DUC benchmark collections are too small for such learning techniques to be effective without the risk of over-fitting. Both these results show an improvement, in terms of ROUGE Score, over various baseline techniques like Round Robin selection, Borda count and a few other techniques.

The work by [32] is perhaps most relevant to the proposed method. They select four well-known summarisation systems (one of which is the best performing system at TAC-08 and TAC-09 w.r.t ROUGE-2) and compute the empirical upper bound on ROUGE-1 and ROUGE-2 recall by examining all possible combinations of sentences, limiting each combination to approximately 100 words. They further define a variety, of features which are then used to train a Support Vector Machine. This is an interesting approach, but since the aggregate summary has to be generated from the given summaries only and no other sentence from the document can be included there is a limit to which it can improve the performance. This will be achieved when the best combination of sentences are selected from given candidate summaries, and no further improvement is possible. The number of possible combinations of sentences that are required for training increase exponentially with increase in the number of candidate summaries. Also compared to [64], where only ranks from various candidates systems were used as features, features extracted from the text are used in this approach which makes the trained system genre specific (DUC dataset contains only news articles). In case a different genre of documents is to be summarised using this system a new training set would have to be created, which is a significant limitation.

In the techniques mentioned above sentences are first ranked independently, and then the rank lists from various systems are combined to form the aggregate ranking. In contrast, there are a few techniques that attempt to aggregate summarisation techniques at the system level. In such cases, the system itself depends on inputs from multiple perspectives. For example, in [55] several sentence similarity measures are combined, and the aggregate score is used with submodular optimisation based technique mentioned in [42]. They show that aggregation of two to four different sentence similarity measure results in improved performance compared to individual similarity scores. The authors combine a sentiment similarity score with common measures like tf-idf or word overlap to generate an aggregate score. One drawback of this method is that it might not always be possible to normalise such different similarity metrics so that they can be combined meaningfully. Instead, we propose using all the similarity metric separately and then fuse the rank lists generated by these different systems. Another approach to aggregate information or features from multiple sources has been suggested by [23], where they propose a framework to integrate features of various granularities like sentence level, document level, collection-based features and features based on other related documents, to identify essential *catchphrases* in a legal document. These catchphrases are then used to generate summaries. The paper by [76] adopts a similar approach by using fuzzy logic to combine features at various granularities. Such approaches are complementary to the proposed approach. Unlike the work by [23] or [76], which focus on aggregating various features, the proposed approach depends on the individual summarisation systems to incorporate those features and then combines the final rank lists from these individual systems.

The first system proposed by us looks into combining several sentence similarity scores to generate a more robust summary[46, 48]. We show that using various combinations of ranking algorithms and sentence similarity metrics generally outperforms individual systems. The second system we propose looks at weighing each candidate system based on their *expected performance*. The approaches by [64] and [80] have a similar aim. Those approaches focus on combining the sentence rankings from candidate systems using weighted linear combinations. While the former relies on a supervised approach that uses SVM-rank to learn relative rankings for all sentence pairs, the latter uses an unsu-

pervised approach based on consensus between the candidate rankings. Existing summarisation datasets are too small to train a generic supervised model. In this thesis, we focus on consensus-based methods to generate aggregates. While similar in principle to Weighted consensus summarisation (WCS)[80], our approach differs in the way in which we define consensus. Unlike WCS, we do not consider sentence rankings to compare two systems. Rather we analyse the overlap in content selected by these systems to measure the consensus between them. We also take into account the relative performance of these systems for individual documents, thus ensuring that best performing system gets more weight compared to the ones with weaker performance.

2.3 Sentence compression

Sentence compression techniques form the core of the majority of abstractive summarisation techniques. Most abstractive techniques work in two parts, sentence extraction followed by sentence compression or abstract generation. Traditionally, a majority of sentence compression techniques have relied on linguistic resources. [11] use tree transduction based rules on aligned and parsed sentences to generate compressions. The work by [81] focuses on the use of quasi-synchronous grammar for summary generation. A noisy channel model was proposed by [16], which consists of a source model or the language model, channel model which estimates the extent to which the original sentence is a good expansion of the compressed sentence and the decoder which searches all possible sentence combinations for a summary. [10] treat sentence compression as an integer linear programming problem. [3] proposed using statistical machine translation for generating news headlines. However, that approach was limited to generating headlines for news articles, and the target sentence was on an average five words long. We use this model as a baseline in our experiments. We show that phrase-based translation is quite efficient in sentence compression, and under certain constraints, it can also work with much longer sentences like those in legal documents.

With the success of neural networks in several NLP applications, data-driven techniques have been immensely popular. There have been several attempts using such techniques for sentence compression. The work in [22] focuses on creating a parallel corpus

of sentence-compression pairs. They propose using the lead sentence in an article as the source sentence and the article headline as the target compression. They use tree pruning on the original sentence to generate the sentence compression. Such a method depends on several resources like parser, pos tagger, NE identification and anaphora resolution. They further use this dataset to train an LSTM based sentence compression system [21]. In this case, the problem is treated as a sequence labelling problem, with each word being labelled as *important* or *not important*. The techniques proposed in [9, 68] use attention networks to generate news headlines. While [9] uses RNNs for sentence encoding, [68] uses CNN for the same. This model does not take into account document level context, which can play an important role in identifying important phrases. For example, a sentence having the name of *appealant* or *respondent* would likely be crucial. The models in [68, 9] does not take into account such context. [8] propose a combined model to extract sentences and further compress the sentences by extracting important words and phrases. This model is in principle similar to the overall model proposed in chapter 6. They use 200K documents and more than 500K sentence-headline pairs from the daily mail corpus as training. The method used for alignment of source sentence to target sentence is similar to what we use in this thesis. The model uses LSTMs for document encoding which requires a much larger training set to be effective. Instead, we use topic models for encoding the document level context and pass it as a topic vector to the decoder. Several works focus on ways to deal with unknown words, [28, 27] in neural network based models. Given the nature of our corpus, the unknown words largely consist of named entities related to the respondent, appellant, jury or witnesses. We use the meta information associated with each case, to replace these named entities with placeholders. Alternately in the final experiment of this thesis, we provide such rare words as a separate context vector to the sentence compression module. A detailed analysis of several states of the art sentence compression techniques is presented in the sixth chapter, where we discuss the pros and cons of these systems and compare their performance.

CHAPTER 3

Domain specific extractive summarization

Automatic text summarisation, especially sentence extraction, has received a great deal of attention from researchers. However, a majority of the work focuses on newswire summarisation where the goal is to generate headlines or short summaries from a single news article or a cluster of related news articles. One primary reason for this is the fact that most public datasets related to text summarisation consist of newswire articles. Whether it is the traditional DUC or TAC datasets or the newly introduced CNN/Daily mail corpus, the focus is mainly on newswire articles. In reality, this forms a rather small part of the numerous possible applications of text summarisation. The focus is now shifting towards other areas like product-review summarisation, domain-specific summarisation and real-time summarisation. Each of these areas has their own sets of challenges, but they have one issue in common, i.e. availability of large-scale corpora which can be used for supervised or semi-supervised learning.

In this thesis, we highlight two such use cases, related to summarising legal and scientific articles, which are very different from the generic document summarisation tasks. To begin with, we introduce two new corpora for document summarisation, which are amongst the largest openly available datasets for abstractive summarisation. With the increasing use of deep learning techniques for solving various NLP problems, large volumes of training data are more critical than ever before, but few of them are public[68] [21]. The proposed datasets, though not as large as the CNN/Dailymail corpus, are the large enough to be useful in training deep learning models. The nature of summaries in both of these cases is such that it is possible to generate useful summaries by merely extracting words and sentences without having to generate new sentences. This is especially important since

progress in sentence generation is limited to date, and most current data-driven approaches use extraction and compression for summarisation. The summaries in case of legal corpora are usually formed by a combination of sentence extraction, keyphrase extraction and sentence simplification. The summaries are recall-oriented, which means it is important to cover as much information as possible at the cost of longer summaries. Summaries, in this case, are $\sim$ 11% of the original documents, which is quite large compared to a summary size of $\sim$ 3% in case of multi-document summarisation in DUC dataset. In contrast, the abstracts of scientific articles are more precise covering only the major highlights of the paper. The proposed approach decides for each sentence, whether or not to include the sentence in summary. This inherently decides the size of summary, which in both cases can vary to a large extent. In the next section, we describe the proposed corpora of legal and scientific articles. Next, we describe several approaches for key-phrase based summarisation for supreme court judgments, followed by a description of the proposed neural model. We compare our results to several existing sentence extraction algorithms.

Overall the major contributions from this chapter are the following:

- Two large scale datasets of Legal and Scientific articles with associated summaries
- Unsupervised techniques that incorporate domain knowledge for sentence extraction
- Completely data-driven approach using weak supervision for sentence extraction and compression

3.1 Corpora

As a part of this thesis, we introduce two new datasets for domain-specific summarisation. The first corpus, which is related to the legal domain, consists of judgements delivered by the *Supreme Court of India*, during the period 1950-1989. Each document has a corresponding summary which was manually written by legal experts and is commonly known as the *headnote* of the judgement. The corpus consists of $\sim$10,000 judgments.

The judgements themselves are publicly available from the website of the Supreme Court of India[]. We tagged these Judgements with the <$INFO$>, <$HEADNOTE$> and

<*JUDGEMENT*> tags. The tagging was done in a bootstrapped manner with successive rule-based automatic tagging followed by manual corrections of a sample of tagged documents. The *INFO* tag primarily consists of information about the case, like the names of the judges who delivered the judgements, names of the petitioner and respondents, those of the prosecution and defence lawyers and the list of other cases that this particular judgement cites. As the name suggests, the *JUDGEMENT* tag contains the detailed judgement that was delivered by the court. Overall it contains the premise of the petition or case, a summary of the facts and arguments, the overall judgement and additional comments, if any, from the bench of judges that delivered the verdict. The *HEADNOTE* tag contains a summary of the corresponding judgement written by a legal expert.

Compared to a general article, e.g. a newswire text or Wikipedia page, legal documents tend to have much longer sentences as well as many abbreviations. We compare statistics from the proposed legal corpus to that of combined DUC dataset (2002, 2003 and 2004). This is shown in table 3.1 below. We also include the statistics of ACL anthology corpus, which we discuss later in this chapter. The number of abbreviations in a legal document is much more as compared to a regular newswire text. This proved to be a bottleneck in sentence tokenisation, with all existing sentence tokenisers performing poorly. To remedy this, we trained an in-house sentence tokeniser that can better handle these abbreviations. Both the judgement as well as the headnote in the proposed corpus are sentence tokenised A subset of this corpus was used in the *Information Access in Legal Domain*, track offered at FIRE 2014 and FIRE 2015.

Table 3.1: Corpus Statistics

		Legal		DUC		ACL	
		Word	Sentence	Word	Sentence	Word	Sentence
Doc Size	Max	90601	763	11754	500	14720	2425
	Min	511	9	1728	92	16	15
	Avg	5500	50	5766	76	4173	201
Summ Size	Max	3234	55	113	12	428	309
	Min	76	1	92	3	7	1
	Avg	500	10	101	6.5	131.73	5.6

Further, we create a sentence level alignment between sentences in the judgements and those in the headnotes. Given the nature of the data, we assume that each sentence in the headnote has exactly one source sentence in the judgement. We further restrict the possible mappings so that the sentences in headnotes are in the same order as those in the judgements. Given a sentence in the headnote(s_h1) and corresponding sentence in the judgement(s_j), the next sentence in headnote(s_h2) can be aligned only to the sentences in the judgement which appear after s_j. This alignment procedure is shown in the pseudo code below.

```
1:  procedure ALIGN
2:      S_h : Sentences in Headnotes
3:      S_j : Sentences in Judgment
4:      k = 0
5:      for i = 1; i ≤ S_h; i++ do
6:          s_max = 0
7:          for j = k; j ≤ S_j; j++ do
8:              if sim(s_i, s_j) > T then
9:                  if sim(s_i, s_j) > sim(s_max, s_j) then
10:                     s_max = j
11:         if S_max > 0 then
12:             k = S_max
13:             Align[i] = k
14:         else
15:             Align[i] = 0
```

The second corpus we propose is, in fact, a subset of the ACL Anthology corpus which

is a collection of scientific articles broadly from computational linguistics and related domains. These articles are openly available in the ACL anthology website[1] in pdf formats. We used the publicly available Science Parse library[2] for extracting section wise information from the pdf documents. Only the articles published in or after the year 2000 were included. Further, the articles that were not parsed correctly were discarded. Finally, 27,801 articles were used in this experiment. We removed the sentences that contain mathematical expressions. This also inherently removes most tabular information. Further, we discarded words with less than three characters. No other pre-processing was used.

As opposed to the legal documents, we use a slightly different method for creating sentence level alignment for the ACL corpus. We use only cosine similarity as opposed to an ensemble of similarity metrics in the former case. This is because usually the sentences are rewritten in case of scientific articles so using n-gram matching or LCS does not provide any particular advantage. For each sentence in the document, we assign a pseudo-label of 1(*important*) or 0(*not important*), based on their cosine similarity with the sentences in abstract. For each sentence in the abstract, we select the best matching sentence from the document if the cosine similarity is above 0.75 (empirically selected) and assign it a label 1. All other sentences are assigned a label 0. In the next section, we discuss in detail the proposed model for sentence extraction.

3.2 Legal document summarization

In this thesis we propose a novel sentence extraction technique as a first step towards automatically generating the headnotes of supreme court judgements. We demonstrate that a very efficient sentence extractor can be created using this data, with weakly supervised training and without any manual labelling. The main contributions are twofold; firstly we propose a simple context encoder which can capture the overall theme of the document and generate a context embedding. Second, we propose an attention model that uses sequence encoder based sentence embeddings along with this context embedding to assign importance to different words in a given sentence. This module jointly learns to capture the

[1] http://aclweb.org/anthology/
[2] https://github.com/allenai/science-parse

informative content as well as the cue phrase information that make a sentence summary worthy. As we show in the results, our model is able identity and leverage cue phrases information to decide the *summary worthiness* of a sentence. Using this information, we can maintain the overall structure of document when creating the headnote, which is not possible using the existing extractive techniques. Contrary to most of the existing techniques, our approach is not dependent on manually tagged data or any linguistic resources.

3.2.1 Boosting legal vocabulary using a lexicon

The similarity between sentences plays a crucial role in most unsupervised extractive techniques. When computing similarity of two sentences, not each word would be equally important. It is possible for a sentence to have a large vocabulary overlap with another sentence, but have only a few *important* terms in common. Two sentences with a couple of informative terms or phrases in common can sometimes share more information compared to two sentences which overlap in more number of commonly used terms. This *informativeness* of terms can be defined in several ways, for example, tf-idf scores, but for a domain-specific task, a term can generally be considered to be informative when it is related to that particular domain. For our first baseline we propose the LegalBoost system. We use relative entropy of the terms for measuring whether they are specific to a legal corpus or not. We build a lexicon of legal vocabulary in the following manner. We first select the top 20% of words occurring in the FIRE legal corpus (all 1500 documents) based on higher tf-idf score. Similarly, we select top 20% terms from FIRE 2011 English ad-hoc retrieval corpus[63]. We then compare the frequency of these terms in the two corpora using the Kullback Leibler divergence. Based on *KLD* between their frequencies in the two corpora, we select the top 15,000 words to constitute the lexicon. We present a sample of this lexicon in Appendix A.

These corpus specific terms are then given more weights compared to other terms when computing similarity. For example, the similarity measure used by TextRank then becomes:

$$S(s_1, s_2) = \sum_{w_1 \in s_1} \sum_{w_2 \in s_2} b_{w1} * (w_1 = w_2) \tag{3.1}$$

Similarly LexRank would be modified to use a soft cosine measure with similar b_w weights.

$$S(s_1, s_2) = \frac{\sum_{i,j}^{N} b_w s_{1i} s_{2j}}{\sqrt{\sum_{i,j}^{N} b_w s_{1i} s_{1j}} \sqrt{\sum_{i,j}^{N} b_w s_{2i} s_{2j}}} \tag{3.2}$$

$$b_w = 1.25 \text{ if } w1 \ \& \ w2 \text{ are corpus specific}$$
$$= 1 \text{ otherwise}$$

The technique Topicsum, discussed in chapter 2, relies on such informative terms. The original TopicSum method computes this using Chi-square statistic. As shown in the table 3.2 below, the proposed lexicon-based method marginally outperforms the original Topicsum approach.

3.2.2 Weighted TextRank and LexRank

All existing extractive techniques in their original form consider only a binary similarity between a pair of words. In case of legal documents, often lexical similarity is not sufficient, and there can be various degrees of similarity between two pairs of words. We try to capture that variation in similarity using the wordnet and observe its effect on the overall ranking of sentences and hence on the quality of the summary. We use wordnet based similarity to capture syntactic similarity as opposed to the standard lexical similarity between sentences. We define the following modified TextRank sentence similarity score for factoring in the degree of similarity between two words:

$$S(s_1, s_2) = \sum_{w_1 \in s_1} \sum_{w_2 \in s_2} path\_sim(w_1, w_2) \tag{3.3}$$

path_sim refers to a score denoting how similar two-word senses are, based on the shortest path that connects the senses in the is-a (hypernym/hyponym) taxonomy as specified by wordnet [53]. Similarly, we can again modify cosine sentence similarity to a

soft-cosine similarity and include $path\_sim$ scores as weights.

$$S(s_1, s_2) = \frac{\sum_{i,j}^{N} b_{ij} s_{1i} s_{2j}}{\sqrt{\sum_{i,j}^{N} b_{ij} s_{1i} s_{1j}} \sqrt{\sum_{i,j}^{N} b_{ij} s_{2i} s_{2j}}} \tag{3.4}$$

$b_{ij} = path\_sim(s_i, s_j)$, s_{1i} refers to word i in sentence s_1

As we show later in the results, this approach improved the performance substantially, but at the same time, it was too slow to be practically useful. Nonetheless, we use this as a competitive baseline.

3.2.3 Automatic key phrase identification

Summaries of legal judgements are a peculiar case in the sense that they are semi-abstractive. Generally, entire sentences are picked up from the judgements and then optionally simplified or compressed by deleting certain phrases or replacing them with simpler vocabulary. In either case, the decision about which sentences to pick can be made based on whether or not they capture certain specific types of information. It is possible to establish the summary worthiness of a sentence based on whether or not it contains certain keyphrases. As opposed to scientific articles, where common phrases like "In this work we describe..." or section information are key features, in case of legal articles the key phrases depend on the type of case. Similar cases have similar key phrases. Keeping this in mind, in case of legal documents we also compare the proposed approach to unsupervised keyword extraction approaches. We use these unsupervised techniques for keyword extraction and then use this in a setup similar to the topic signature based technique. We employ two existing unsupervised approaches for identification of key phrases and then further use this information to assign weights to the individual sentences. We use YAKE [6] which uses an ensemble of various rules which are used to identify important phrases in a given piece of text. The technique is agnostic to domain or size of the articles. We employ this is a setup similar to that of topic signatures based technique[41], where each sentence is scored based on the number of key-phrases it contains. Alternately, we use the approach suggested in [45], where they focus on extracting *catch-phrases* from legal documents. In

this work, the authors define various rules for extracting important noun phrases. These are then ranked in the order of their importance using several heuristics. In case of YAKE, the code is available as a python library, and we need to specify input text. In the latter case, we contacted the authors, who provided us with a list of catch-phrases for the judgements which we use directly. In both cases, the main idea is the same as that in the topic signature based method. Only the definition of *key phrases* changes. Both the methods prove to be an improvement in over the original topic signature based technique. The results are compared in table 3.2.

3.2.4 Attention based sentence extractor

Our proposed model consists of four main blocks: A lstm based sentence encoder, topic modelling based context encoder, attention module and a binary classifier. Overall the aim is to determine the probability $p(y|s,d)$ where $p(y)$ is the probability that sentence s in a document d is *summary worthy*. We represent s as an embedding vector of fixed dimensions, using a sentence encoder. Next, we represent each document by the topic extracted using LDA, and use those topics to create a context embedding. The attention model then uses the sentence and context embeddings to learn to assign weights to different parts of the sentence. Finally, the classifier uses the output of attention module and the original context embeddings to decide whether or not a sentence is *summary worthy*. Below we describe the individual blocks.

Sentence encoder

Each sentence S is represented as a sequence of N vectors $[\boldsymbol{x}_1,...\boldsymbol{x}_N]$ where $\boldsymbol{x}_i$ is the i^{th} word represented by its word embedding vector. The initial word embeddings were created by training a word2vec[51] model on a corpus of 300K legal documents available from FIRE Legal Track[3], and were updated during the training. The word embedding matrix $\boldsymbol{E}$ is of size $V \times D$, where V is the vocabulary size, and D is the word embedding size. Next, we use an LSTM based sequence encoder with a hidden size of U for creating the sentence embeddings using these word embeddings. LSTM based sentence encoders are now considered a standard technique for creating sentence embeddings. We limit the

[3] fire.irsi.res.in

maximum possible length of a sequence to L.

Context encoder

Even for humans, knowledge about the overall scope of an article is pivotal when selecting important information that has to be included in the abstract. There have been attempts to generate a document encoding, by using an additional LSTM based sequence encoder that takes input a sequence of sentence embeddings created by the sentence encoder defined above and gives a single vector or the *document embedding*[8]. However, such an approach requires a significant amount of training data, of the order of hundreds of thousands of article, and takes much longer to train. As an alternative, we propose a more straightforward approach, which efficiently captures the overall scope of the document and can be trained using only a few thousand documents. It is noteworthy that here our aim is not to capture the document structure explicitly but to capture the overall theme of the document.

Our context encoder follows a two-step approach. In the first step, we encode each judgement in the form of representative concepts present in them. We extracted 100 abstract topics from the overall corpus using *Latent Dirichlet Allocation* based topic modelling. Topic vectors for each document can be represented as a matrix $T \in \mathbb{R}^{M \times M}$, $\boldsymbol{T}$ = $[t_1, ..., t_M]$, where t_i is the one-hot encoded vector of size $1 \times M$ for topic i, and M is the pre-decided number of topics. We separately initialised a topic embedding matrix $\boldsymbol{F} \in \mathbb{R}^{M \times C}$, where M is the total number of topics and C is the context embedding size. We randomly initialise $\boldsymbol{F}$, and it is jointly updated with the overall model. $J \in \mathbb{R}^{C \times M}$ represents the topic embeddings. We then perform a weighted average of the topic embeddings using their probabilities($\boldsymbol{p_i}$). This additional step helps in reducing the sparsity of LDA representation as well as to leverage latent similarity between different topics while at the same time assigning an importance score to each of the topics. $c \in \mathbb{R}^{C \times 1}$ represents the final weighted context embeddings.

$$J = F^{\intercal} T \tag{3.5}$$

$$c = \sum_i p_i J_i \tag{3.6}$$

Attention module

This module plays a key role in the overall architecture. It specifically learns to identify the *key phrases* in a sentence. In past few years, attention networks have become a popular choice for several NLP tasks. Several approaches have been proposed for using attention for document summarization[8],[68]. We propose a simple attention architecture that takes into account the document context and sentence embedding for generating attention weights over the sentence words. We argue that besides identifying informative content in the document, such an attention model would help in automatically identifying words or phrases, which can act as a cue for deciding whether or not that sentence is summary worthy. The attention weights($[w_1, ..., w_L]$) are computed as shown in equation 3, where $Z \in \mathbb{R}^{(S+C)\times L}$ and $w \in \mathbb{R}^{L\times 1}$. The attention module learns weights $\boldsymbol{w}$ as a function of the sentence embedding(local context) as well as the context embedding (global context). L is the maximum allowed length of an input sentence. Sentences shorter than this are padded to make them of the same length. $Y \in \mathbb{R}^{L\times S}$ denotes the intermediate steps of LSTM output at each of the L timestamps. $Y = [y_i, ...y_L]$ where $\boldsymbol{y_i}$ represents intermediate output at a particular time stamp i.

$$\boldsymbol{w} = Z(s, c) \tag{3.7}$$

$$\boldsymbol{a} = w^{\mathsf{T}} Y \tag{3.8}$$

Classifier

The classifier consists of two layered feed forward network. We used a hidden layer with weights $\boldsymbol{H} \in \mathbb{R}^{(A+C)\times Q}$ followed by a output layer $\boldsymbol{O} \in \mathbb{R}^{Q\times 1}$ and a sigmoid activation function (σ).

$$h = \boldsymbol{H}[a, c] \tag{3.9}$$

$$o = \sigma(\boldsymbol{O}h) \tag{3.10}$$

The entire architecture is shown in the Figure 3.1 below.

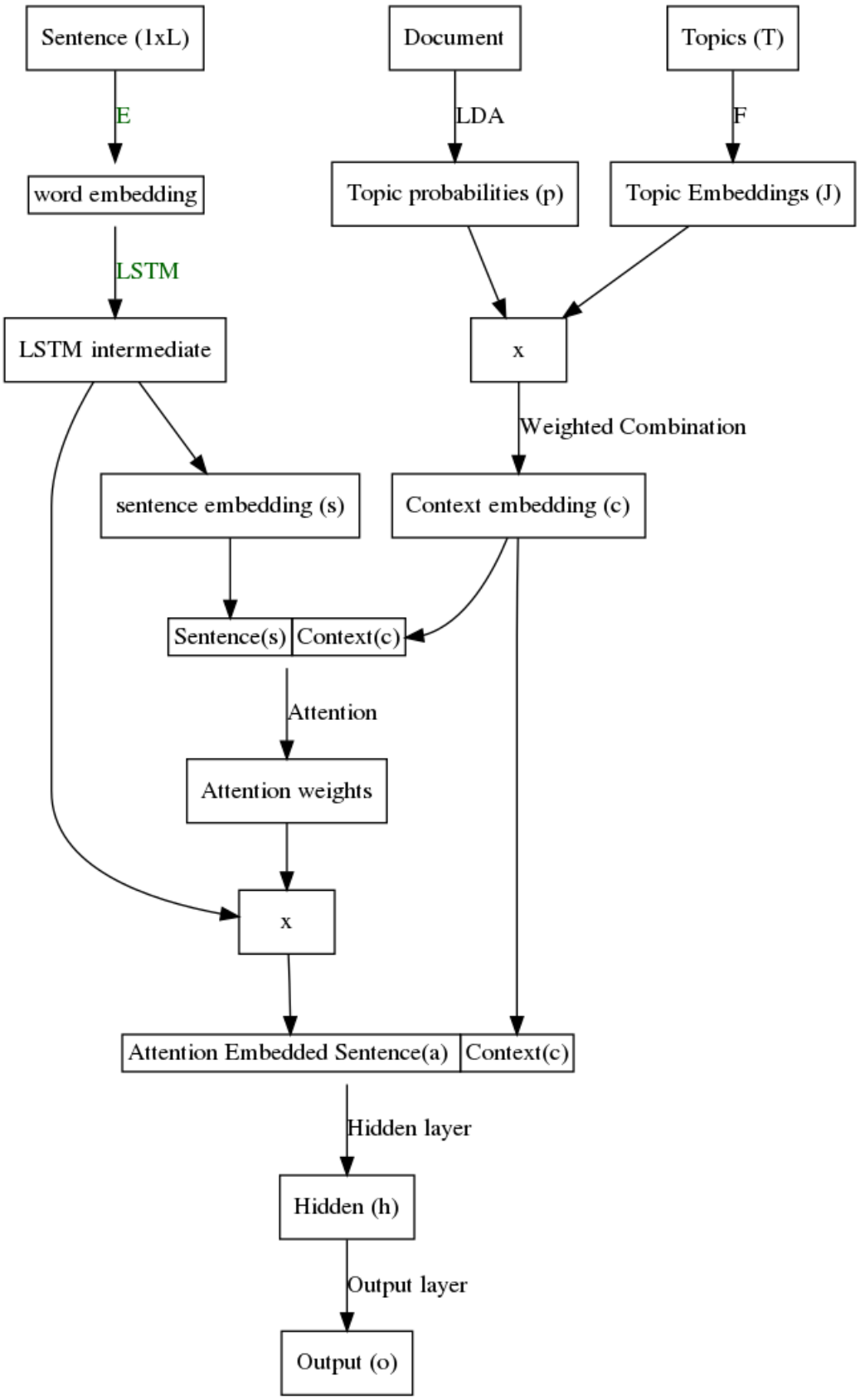

Figure 3.1: Attention based sentence selector

3.3 Scientific article summarization

Scientific article summarisation has often been attempted from diverse perspectives like capturing novelty in the proposed work, using citation graphs to identify relevant content or use rhetorical role labelling for creating summaries. As opposed to the data-driven nature of the proposed approach, all the previous experiments using scientific articles were limited to not more than a few hundred articles[1]. Our approach is different for several reasons. One, it does not rely on any linguistic or domain knowledge thus making it more robust to change in the domain of scientific articles. Second and more important, it automatically learns the cue-phrases. These phrases play a pivotal role even when humans are generating the abstract, and all previous attempts define a list of such cues as compare to our approach, which identifies them automatically from the data. The most notable attempt at summarising a scientific article was by [78]. They solve this problem by leveraging rhetorical status of sentences to generate the abstract. The idea is to select sentences in such a manner that the abstract highlights new contribution of the paper and also relates it to the existing work. The authors identify *Rhetorical zones* to which a sentence can belong like, the aim of the experiment, statements that describe the structure of the article, comparison with other works, etc. Such features are then used to decide the importance of the sentence as well as to maintain the overall structure of the final extract. The work by [49] focuses on generating impact based summaries for scientific articles. Sentences are ranked based on the impact they have produced on other works in the same or related domains. *Document sentences* that best match the content of the *citing sentence* are identified using language models. They used a dataset of around 1300 papers published by ACM SIGIR. The work described in [1] clusters articles based on their citation graph and then use lexank to rank sentences within each cluster. The work proposed in [30] focuses on identifying the sections of the original paper to which a given abstract sentence is related.

Compared to a summary of a newswire cluster, abstracts of scientific articles are much

more precise with a higher compression ratio. The average size of ACL articles is 200 sentences or about 3600 words, while the average summary size is 125 words. This poses a challenge which is different from that in the legal corpus. The abstracts are precision oriented, as opposed to the recall-oriented summaries in legal documents. Nonetheless, our model is robust enough to handle this difference. It individually rates sentences, and we can then select the top-k sentences where the value of k is domain dependent. Cue phrases play a pivotal role in this case. For example the sentences with known cue phrases like, "In this work, we propose..", or "We conclude that.." get generally higher weights, and the attention model very accurately captures such phrases. The nature of data, where very few sentences are considered to be summary worthy results in a heavily skewed training data, with more than 95% sentences being labelled as *not important*. To mitigate this bias, Apart from this we also subsample the negative examples from training data. First, we filter out sentences with tf-idf scores lower than 0.05 as these sentences generally contain non-informative content. Further, we randomly sample the *not significant* sentences to bring down the positive-negative ratio to 1:4. We then use a weighted loss function, to assign a higher loss to false negatives as compared to false positives. We use the weighted binary cross-entropy loss to mitigate the class imbalance issue partially. We use a weight of 0.2 for negative samples and 0.8 for positive samples.

Our model implicitly tries to learn similar information. In the results, we show that the attention model learns to identify phrases which are indicative of the section information and such sentences are usually selected in the summary. Another related work to the proposed approach is by [8]. It focuses on generating headlines of news articles using sentence and document encoders. The authors use sentence and word level labelling to identify essential phrases in the document and then generate an extract based on that technique.

3.4 Experiment details

We used the pytorch library[4] for our experiments. For Adam optimizer we use the most common setting with a learning rate of 0.001, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Training was performed with shuffled mini batches of size 500, and a dropout of 0.2 was used for all layers. All the random initialisations used Xavier normal distribution. We used $D = 100$(word embedding size), $C = 10$(context embedding size) and $M = 500$(number of topics). We used a single LSTM layer with 200 hidden states and Q = 100(classifier hidden layer size). We plan to make the source code publicly available.

We evaluate the performance of our system on a held out evaluation set, 700 documents in the legal corpus and 2800 documents in the ACL corpus. ROUGE metrics[40] is used to compare the system generated extracts with the original abstracts of the papers. We report ROUGE-N Recall for the legal corpus (N = 1-4). In contrast to legal document summarisation, summarising scientific documents is a precision-oriented task, and hence We report ROUGE-N precision (N=1,2,4) in this case. We compare our results with five widely accepted extractive summarisation systems besides two state or art techniques. We specifically choose the topic signature[41] and latent semantic analysis[75] based approaches due to their ability to identify the overall context and latent topics in a document. Besides these, we also compare our results with the popular graph-based approaches, lexrank[20] and textrank[50] and a simple frequency-based approach. We also compare the results with Submodular optimization based technique[43] and Integer linear programming based summarization[24], which are considered to be state of art techniques for sentence extraction[31]. Additionally, in case of legal corpus we compare the results with other phrase based techniques like LegalBoost, weighted TextRank, weighted LexRank, YAKE and catch-phrase based summarizer explained earlier.

To make the results reproducible, we follow the guidelines suggested in [31] and use a fixed set of parameters when computing ROUGE scores[5]. The value Z depends on the target summary. In general, for experiments on the standard DUC corpus, the summary limit is 100 words. However, in our case, this can vary significantly across domains. Since

[4] http://pytorch.org/

[5] ROUGE-1.5.5 with the parameters: -n 4 -m -a -l Z -x -c 95 -r 1000 -f A -p 0.5 -t 0

the summary size varies across documents in both the evaluation sets, we use the average summary length when computing the ROUGE scores. The summary length (value Z in the Rouge parameters) is 125 words for abstracts of ACL articles, and 500 words for the legal corpus. All other parameters are same as those mentioned in [31].

3.4.1 Results

As evident from table 3.2, the proposed approach performs very well on the legal dataset. It achieves a substantial improvement compared to all the baselines. Even compared to the methods using automatic keyphrase detection, our approach performs significantly better. The difference is less for Rouge-1 and Rouge-2, but the improvement on R-3 and R-4 is substantial. We see similar trends for the results on ACL corpus as well. We also discuss, later in this section, the reason for this difference when using higher level Rouge scores compared to R-1 and R2. The figures in bold in following tables indicate the best performing system for a given rouge metric. A † sign indicates a significant difference between the best performing system and the next best system. We used a two-sided sign test with $\alpha - 0.5$ for all comparisons.

Table 3.2: Results on Legal Corpus (ROUGE-N Recall)

Summarizer	R-1	R-2	R-3	R-4
LSA	0.630	0.328	0.210	0.142
LexRank	0.658	0.350	0.240	0.155
TextRank	0.643	0.342	0.229	0.160
FreqSum	0.629	0.344	0.238	0.160
Submodular	0.695	0.384	0.265	0.175
ILP	0.678	0.371	0.262	0.180
Topicsum	0.670	0.355	0.260	0.178
YAKE	0.675	0.380	0.243	0.159
LCP	0.682	0.390	0.259	0.181
LegalBoost	0.662	0.367	0.240	0.150
TexRank-W	0.682	0.360	0.257	0.155
LexRank-W	0.684	0.364	0.248	0.135
Neural	**0.715**†	**0.415**†	**0.279**†	**0.215**†

Figures in bold indicate the best performing system

† indicates significant difference with $\alpha = 0.05$

The results on ACL corpus are shown in table 3.3. As evident, the proposed approach outperforms the existing systems on most ROUGE metrics. The only exception is Rouge-1 measure, where Submodular performs the best. We observe that R-3 and R-4 better reflect the systems ability to retain structural information in the abstract. A summary with good R-1 or R-2 has more informative phrases but misses out on the structural information. A summary with higher R-3 or R-4 usually prefers sentences with certain cue phrases like 'results are significantly higher compared to' or 'in this paper we propose'. This is closer to the way a human would decide whether or not to include the information.

Table 3.3: Results on ACL Corpus (ROUGE-N Precision)

Summarizer	R-1	R-2	R-3	R-4
Topicsum	0.266	0.055	0.020	0.012
LSA	0.302	0.065	0.027	0.018
LexRank	0.354	0.087	0.037	0.020
TextRank	0.305	0.074	0.030	0.018
FreqSum	0.331	0.088	0.034	0.018
Submodular	**0.360**	0.087	0.036	0.022
ILP	0.350	0.082	0.0355	0.021
Neural	0.344	**0.090**	**0.042**†	**0.027**†

Figures in bold indicate the best performing system

† indicates significant difference with $\alpha = 0.05$

Below, we include a summary generated by our system along with the original abstract of the paper. The legal headnotes are too long to include here. Instead, the results on legal documents are included in the appendix along with additional results on the ACL corpus. The intensity of highlight shows the attention module assigned higher weights to those phrases. Darker the shade, higher the attention. In general, we observe that the proposed attention model efficiently identifies content words and cue phrase, both of which are important when selecting a sentence. For example, consider the first sentence of system generated summary: "In this paper we propose a statistical model for measure word generation for English-to-Chinese SMT systems, in which contextual knowledge from both source and target sentences is involved.". Our model identifies the phrases "In this paper, we propose" (cue phrase) and "statistical model for measure word" (part of the title) as important.

Document ID: P08-1011

Original Abstract: Measure words in Chinese are used to indicate the count of nouns. Conventional statistical machine translation (SMT) systems do not perform well on measure word generation due to data sparseness and the potential long distance dependency between measure words and their corresponding head words. In this paper, we propose a statistical model to generate appropriate measure words of nouns for an English-to-Chinese SMT system. We model the probability of measure word generation by utilizing lexical and syntactic knowledge from both source and target sentences. Our model works as a post-processing procedure over output of statistical machine translation systems, and can work with any SMT system. Experimental results show our method can achieve high precision and recall in measure word generation.

System generated summary: In this paper we propose a statistical model for measure word generation for English-to-Chinese SMT systems, in which contextual knowledge from both source and target sentences is involved. To overcome the disadvantage of measure word generation in a general SMT system, this paper proposes a dedicated statistical model to generate measure words for English-to-Chinese translation. Experimental results show our method can significantly improve the quality of measure word generation. We also compared our method with a well known rule-based machine translation system - SYSTRAN3. Most existing rule-based English-to-Chinese MT systems have a dedicated module handling measure word generation.

Sample abstract and system generated summary

It is also interesting to note that the proposed model efficiently captures overall structure of the document, it starts with proposed work, then some details about experiment and system comparison. Barring the last sentence, it is quite precise and coherent in terms of content. Although it is not always possible to have sentences in the original documents that can directly be included in the abstract, the results of the current experiment are quite encouraging and can serve as an excellent first step towards abstract generation.

3.5 Conclusion

In this chapter, we proposed a weakly-supervised approach for generating extracts from legal and scientific articles. We use topic models to create a context embedding that defines the scope of the article and then use an attention-based sequence encoder to generate sentence encoding. We then use pseudo labelled data to train a classifier that predicts whether or not a given sentence is *summary worthy*. When evaluating on ACL anthology corpus, we were able to outperform the existing baseline and state of the art techniques on ROUGE-2,3 and 4 metrics, while achieving a comparable performance on ROUGE-1. Moreover, we also demonstrate that our approach well preserves the overall structure of original document resulting in a final summary that is quite coherent. We envision this as a first step towards automatically creating abstracts of legal and scientific articles. Sentence compression techniques can further use the results of this extractive step.

CHAPTER 4

Improving extractive techniques through rank aggregation

Numerous extractive summarisation techniques have been suggested in the past that range from pure frequency-based approaches such as FreqSum[58], that rank sentences based on the frequency of its words in the document, to the sophisticated sentence compression techniques, like [11], which also focus on rewriting sentences besides compression. As a result of the standard evaluation benchmarks generated by conferences like DUC[14] and TAC[60] theoretically, it has become possible to compare such systems on a standard benchmark. However, multiple factors, like pre/post processing and evaluation setup amongst others, that affect the performance of extractive systems, and it is always unclear what role these factors played in the success(or failure) of a particular technique. Besides this, the performance is also susceptible to the parameters used when evaluating with the ROUGE toolkit. The same system can obtain very different ROUGE scores when evaluated with a different set of parameters. These factors make it difficult to say if a particular technique is better than the other or the difference is just because of the bias introduced by these external parameters. It would also be interesting to see, how similar the summaries with similar ROUGE scores are. We present a study which highlights the role of individual components of a summarisation system and proposed a method for leveraging this information to generate better summaries. Overall, in this chapter, we try to answer the following questions

- What effect do the external factors of like stemming, stopword removal, redundancy removal, etc. have on the overall performance?
- How sensitive is a new summarisation system to choice of sentence similarity mea-

sure, ranking algorithm and text representation scheme? Is some particular combination always better?

- Can the answers to above questions be leveraged to generate better ensemble system?

We start by highlighting the effects of preprocessing and post-processing steps on the performance of the overall system. We then proceed to demonstrate the effect of variation in three principal components of an extractive summarisation system:

- Sentence similarity metric. The usual choices are cosine similarity, word overlap and Kullback Leibler divergence.
- A Ranking algorithm is the core of any extractive technique. The most common ways to ranks sentences is using a graph-based approach[50, 20], greedy algorithm[29] or centroid based approach[67].
- Text representation scheme is usually a latent choice that is not explicitly highlighted. The most popular choice is the tf-idf based representation, but there are other representations like topic signatures[41] or latent semantic indexing[19].

Any new extractive summarisation technique is proposed with a particular choice of these components as well as other pre/post-processing steps. However, they usually fail to provide any insights as to why a particular combination was preferred. Most combinations of the available sentence similarity metric and ranking algorithms would be valid and can form a new summarisation system of their own. We use such variations in to create a large number of relatively similar systems and show that the original combinations used in existing techniques are not always the optimal choice.

Since it is not always possible to predict which combination would perform the best, we propose using multiple sentence ranking algorithm/similarity metrics instead of choosing any combination in particular. Unlike the previously reported work on similar lines by [80], we observe that merely combining the ranked lists of sentences from multiple systems does not always produce an effective metasystem. As compared to that, aggregating

ranked lists in an informed manner almost always improves the overall performance. We experiment with the DUC 2003 and DUC 2004 datasets, which have become commonly accepted datasets for evaluating multi-document summarisation tasks. We further describe these datasets in section 4.3.

4.1 Motivation for rank aggregation

Most extractive techniques comprise of three fundamental components: (1) Sentence similarity metric, (2) Sentence ranking algorithm and (3) Text representation scheme. These components are relatively independent of each other and, as we show in our experiments, their choice can drastically alter the performance of a summarisation system. However, few inquiries have been made as to how changing these components affects the overall effectiveness of the system.

In their work, [31] provide a comparison of summaries generated by several standard and state-of-art techniques. They use a common ROUGE setup, to eliminate any variation in performance due to that factor. To be consistent and to make the work in this thesis reproducible, we use the same set of ROUGE parameters as that used by [31][1] in all our experiments. This study highlights the fact that although many different systems have similar ROUGE scores under this common setup, the content across these is substantially different. This indicates that there is a scope for combining summaries generated by different systems and doing so can improve the coverage of resultant summary. Despite this, few attempts have been made to leverage this difference in performance of summarisation systems to create an ensemble or inquire if such an ensemble will be useful at all. In this chapter, we first highlight the effect of various pre/post-processing steps on the overall systems and show that the choices made in the original works were not always the ideal ones. Hereafter, all the existing systems that are used for our experiments will be termed as the *candidate systems*, and the systems generated by combining them as *ensembles* or *meta-systems*. We then propose a strategy to create variations of the existing systems and use these variations to generate a better ensemble, that outperforms all the candidate

[1] ROUGE-1.5.5.pl -n 4 -m -a -x -l 100 -c 95 -r 1000 -f A -p 0.5 -t 0

systems.

4.2 Analysis of existing extractive systems

In this experiment, we try to get some insights about the factors that can affect an extractive summarisation system. Apart from the common pre-processing steps (like stemming, stop word removal, etc.) three major components that are central to most extractive summarisation systems are (1) sentence similarity metric, (2) sentence ranking algorithm and (3) Text representation scheme. Most techniques are proposed with a particular combination of sentence similarity metric, ranking algorithm and text representation scheme but usually fail to provide any insights as to why this particular choice. Most combinations of the available choice of components would be valid and can form a new summarisation system. As a result, when a new technique is proposed it might not always be clear what made a more significant impact, the way sentence similarity is defined, sentence representation, choice of a particular ranking algorithm or merely some other pre/post-processing steps. We investigate the effect of each component in determining the effectiveness of the overall system. This information would not only be beneficial when designing new extractive summarisation systems but can also lead to a better ensemble of existing systems, as we show in the next section.

We choose candidate systems to have representative systems from various categories mentioned in chapter 2. We select both topic representation based systems(Centroid, FreqSum, TsSum) as well as indicator representation based systems(LexRank). These systems cover centrality based techniques, corpus-based techniques as well as graph-based techniques. We aim to highlight the effect of the choice of individual components on the performance rather than an exhaustive study as to what is *the best* approach. The fact that these systems are based on only the three main components, and that no other factors (like pre-processing, use of external resources, etc.) play any pivotal role, makes it possible to highlight the importance of each component. In fact, most state-of-art systems can be traced to one or more of these systems and generally tend to be improved versions of such simpler systems. The candidate systems that we use are briefly discussed below:

- **Greedy-KL**: This technique, proposed by [29] follows a greedy algorithm to minimise the KL Divergence between the original document and resultant summary. Sentences are sequentially added to the summary such that they minimise the KL-Divergence between word distributions of the set of sentences selected so far and the overall document.

- **LexRank**: Each document is treated as a graph, and each sentence in the document constitutes a node[20]. The edges represent cosine similarity between the sentences. The nodes are then ranked based on the Pagerank[62] algorithm, where the importance of a node is iteratively determined by the number of other nodes to which it is connected and in turn the importance of those nodes.

- **Centroid:** A centroid of all sentences is computed and then sentences that are close to the centroid, in terms of cosine similarity, are iteratively chosen[67].

- **FreqSum:** This technique is solely dependent on the ranking algorithm and does not take into account the similarity between two sentences[58]. Sentences having words that are more frequent in the document set are ranked higher.

- **TsSum:** This method is based on extracting representative words, termed as topic signatures, by comparing the text to a large background corpus[41]. The sentences with a higher number of topic signatures are considered more important.

- **LSA:** In this approach the document is represented as a term-sentence matrix which is then projected into the semantic space by using singular value decomposition[75]. The sentences corresponding to top-k singular values are then included in the summary.

The candidate systems used in this chapter include four different sentence ranking algorithms: greedy approach, centroid-based approach, graph-based approach and the frequency based method. Three sentence similarity metrics: KL-Divergence, Cosine similarity and word overlap are used. The last two systems are not used in this experiment but are

used in the next section where we demonstrate the effect of different text representation techniques on the ensemble of summarisation system. The topic signature-based system (TsSum) is similar to the FreqSum based method. The only difference between them is the text representation scheme used (topic signatures vs unigram tokens). We use this fact to highlight the effect of a change in text representation scheme. On the other hand, Latent semantic analysis based method does not use a sentence similarity metric and hence is not included it in the first experiment.

To start with we compare the candidate systems with all possible combinations of sentence similarity measures and ranking algorithms. For example, the original Greedy-KL system used a greedy sentence selection algorithm, and KL Divergence was used for computing similarity between a new candidate sentence and the sentences already included in the summary. However, it is also possible to use some other similarity metric like cosine or word-overlap instead of KLD. Similarly, we can use word overlap in place of cosine similarity in the lexRank system, which in fact was proposed separately by [50]. We create ten different summarisation systems by using various combinations of these two parameters.

The only modification we did in the original algorithms was to use uniform pre-processing techniques for all combinations. In general, the opinion is divided whether pre-processing steps like stop word removal, stemming, etc. improve summarisation results. While a study in[59] suggests using stemming but keep the stop words, [58] perform stemming and remove stop word in the original algorithm, and nothing was explicitly mentioned for several other systems. We observed that for all the systems listed above stemming does not affect the final results significantly, as shown in Table 4.1. On the other hand not removing stop words negatively affect the overall results of all systems except LexRank. For this experiment we used porter stemmer[2] and the standard stop-word list for English[3]. To be consistent, in further experiments we use the same set up across all systems by removing stop words and not performing stemming. We also used a common post-processing step for handling redundancy. A new sentence is selected in

[2] https://tartus.org/martin/PorterStemmer
[3] http://snowball.tartus.org/algorithms/english/stop.txt

the summary only if its cosine similarity with all of the sentences already selected is less than 0.5. This is a commonly used step in literature[31] but was not a part of many candidate systems when they were originally proposed. Table 4.1 below shows the effect that each pre/post-processing step has on the overall result. We also include the result of the best combination, i.e. removing stop words and handling redundancy, but not performing stemming.

Table 4.1: Effect of pre-processing and post-processing steps on ROUGE-1 Recall

	System	No pre/post processing	Only Stemming	Only Stopword Removal	Only Redundancy Removal	Stopword + Redundancy Removal
DUC 2002	Centroid	0.41783	0.42001	0.42223	0.43157	**0.44987**
	Greedy-KL	0.40173	0.40537	0.41392	0.40962	**0.41522**
	LexRank	0.42733	0.42000	0.42292	**0.44134**	0.43289
	FreqSum	0.39247	0.38120	0.40480	0.38766	**0.42522**
DUC 2003	Centroid	0.33387	0.34222	0.34382	0.35237	**0.36780**
	Greedy-KL	0.31473	0.31263	0.33892	0.31592	**0.33892**
	LexRank	0.35643	0.34900	0.34292	**0.36111**	0.35689
	FreqSum	0.29316	0.30120	0.32748	0.30486	**0.34335**
DUC 2004	Centroid	0.35399	0.35104	0.34874	0.36541	**0.37271**
	Greedy-KL	0.31913	0.32215	0.33717	0.31866	**0.34160**
	LexRank	0.35356	0.34343	0.34453	**0.36277**	0.35377
	FreqSum	0.30776	0.31500	0.34816	0.31370	**0.35851**

4.2.1 Experimental setup

For all our experiments we have used the DUC 2002, DUC 2003 and DUC 2004 datasets which have become the standard benchmark corpora for any generic document summarisation tasks. While DUC 2002 contains news articles from TREC collection, both DUC 2003 and DUC 2004 datasets contain clusters of news articles from the TDT (Topic detection and tracking) datasets. DUC 2002 contains 59 clusters of around ten documents each, DUC 2003 dataset contains 30 clusters of about ten documents each and DUC 2004 consists of 50 clusters with ten documents per cluster. All three datasets include four manually written summaries per cluster. We use the ROUGE toolkit[40] for evaluation, and report ROUGE-1 recall, ROUGE-2 recall and ROUGE-4 recall for the first experiment.

As the results for all three measures are strongly co-related, with an average Kendall's tau of 0.85, we report only ROUGE-1 recall for subsequent experiments. Additionally, we report ROUGE-L score for the last experiment, where we vary the text representation scheme.

Table 4.2: Effect of Sentence similarity and Ranking algorithm on ROUGE scores

			Greedy	Graph	Centroid	Freq
DUC 2002	Cosine	R-1	0.418	0.433$^{\diamond}$	0.450$^{*\diamond}$	0.425*
		R-2	0.160	0.225$^{\diamond}$	0.240$^{*\diamond}$	0.232*
		R-4	0.014	0.016	0.019*	0.019*
	Word Overlap	R-1	0.430$^{*\diamond}$	0.422	0.418	0.425*
		R-2	0.235$^{*\diamond}$	0.169	0.229	0.232*
		R-4	0.021$^{\diamond}$	0.017$^{\diamond}$	0.020$^{\diamond}$	0.019*
	KL Divergence	R-1	0.415	0.412	0.422	0.425*
		R-2	0.202	0.162	0.221	0.232*
		R-4	0.016	0.013	0.016	0.019*
DUC 2003	Cosine	R-1	0.337	0.343$^{\diamond}$	0.344$^{*\diamond}$	0.328
		R-2	0.070	0.076$^{\diamond}$	0.081$^{*\diamond}$	0.068
		R-4	0.008	0.008	0.011$^{*\diamond}$	0.009
	Word Overlap	R-1	0.323	0.336*	0.286	0.328
		R-2	0.067	0.070*	0.043	0.068
		R-4	0.010$^{\diamond}$	0.011$^{*\diamond}$	0.004	0.009
	KL Divergence	R-1	0.339$^{*\diamond}$	0.332	0.324	0.328
		R-2	0.074$^{*\diamond}$	0.065	0.066	0.068
		R-4	0.009	0.007	0.010*	0.009
DUC 2004	Cosine	R-1	0.342	0.354$^{\diamond}$	0.373$^{*\diamond}$	0.359
		R-2	0.066	0.075$^{\diamond}$	0.089$^{*\diamond}$	0.081
		R-4	0.008	0.009$^{\diamond}$	0.012*	0.012
	Word Overlap	R-1	0.360$^{*\diamond}$	0.352	0.359	0.359
		R-2	0.082$^{*\diamond}$	0.069	0.082	0.081
		R-4	0.013$^{*\diamond}$	0.009	0.013$^{\diamond}$	0.012
	KL Divergence	R-1	0.342	0.346	0.354	0.359*
		R-2	0.072	0.069	0.078	0.081*
		R-4	0.010	0.008	0.011	0.012*

* indicates the best performing ranking algorithm for a given sentence similarity score, for a given year.

$^{\diamond}$ indicates the best performing sentence similarity score for a given ranking algorithm, for a given year.

Table 4.2 shows the effect of the change in one component keeping the other constant. Since the frequency based technique does not depend on sentence similarity score, the

Table 4.3: Jaccard Co-efficient of bi-gram overlap between summaries (DUC 2004)

	Cosine			Word Overlap			KL Divergence		
	Greedy	Graph	Cent.	Greedy	Graph	Cent.	Greedy	Graph	Cent.
GreedyC	1.000	0.204	0.183	0.250	0.172	0.163	0.190	0.162	0.175
GraphC		1.000	0.183	0.250	0.223	0.169	0.173	0.190	0.211
CentroidC			1.000	0.181	0.175	0.193	0.191	0.178	0.182
GreedyW				1.000	0.179	0.223	0.190	0.176	0.186
GraphW					1.000	0.159	0.177	0.223	0.168
CentroidW						1.000	0.184	0.205	0.210
GreedyK							1.000	0.153	0.178
GraphK								1.000	0.192
CentroidK									1.000

ROUGE scores are constant for a given corpus. We observe that in a few cases new combination performs better than the one that was initially proposed. For example, a greedy algorithm has a significantly higher ROUGE score on DUC 2004 when word overlap is used instead of KL Divergence. Similarly, cosine similarity, when used with the greedy approach, performs equally good as the original greedy and KLD combination. This also holds true for cases where the sentence ranking algorithm is varied keeping the similarity score constantly. The centroid-based technique with cosine similarity performs equally good as the graph-based technique with cosine similarity. However, overall these combinations do not perform significantly different from each other ($\alpha = 0.05$) in terms of ROUGE scores. On the contrary pairwise comparison of these summaries indicate a very low overlap in terms of bigrams ($\sim$0.18 on an average). Table 3 shows the 2-gram overlap, using Jaccard coefficient, between the variants of extractive summarisation systems for the DUC 2004 dataset. These results are similar to those reported by [31] and indicate that although the ROUGE scores of these summaries are comparable, the content across these summaries is very different. This indicates a possibility of building a new summary, with enhanced coverage, by selecting important content from across these systems. Overall these combinations work equally well, and there is no reason to believe one would always outperform the other. This raises an important question: what should be the focus when proposing a new technique? Is it sufficient to improve any one component, while keeping the others constant? We try to answer this in the next section, where we show the effect of aggregating various summarisation systems.

We also report the effect of variation in text representation scheme on the DUC 2004 dataset. The results are reported in Table 4.4. Besides simple TF-IDF we use word2vec, topic signatures and LSA representations. These representations are believed to capture the semantic similarities in words rather than lexical similarities. For obtaining the word2vec representation, we use pre-trained vectors trained using Google News dataset [52]. The model contains 300-dimensional vectors for 3 million words and phrases. A sentence vector is formed by averaging the vectors of all the words. Words missing in the Google news corpus are ignored. Topic signatures are computed as suggested in [41] by comparing the given document to an extensive background corpus. In LSI, the original term-document matrix is reduced to a matrix of smaller dimensions, each dimension representing an abstract concept. We use two sentence similarity scores namely cosine and word overlap for this experiment. Cosine similarity can be easily defined in all four cases as we can directly have sentence vectors in each representation. In TopicSum the sentence vector is similar to that in tf-idf, but it considers the frequency of topic signatures instead of terms. On the contrary word, overlap does not make much sense in case of word2vec or LSA representations, and we use only tf and topic signature representations in that case. Word-overlap in case of topic signature representation is the number of topic common between two sentences, normalised by the total number of topics in both. As highlighted by the results in Table 4.4 the variation in text representation scheme has a significant effect on the overall performance of individual systems. For example, using topic signatures with graph-based technique performs much better than the original Textrank system[50] which used word overlap.

4.3 Ensemble of extractive summarization systems

The results in the previous section clearly show that no particular combination of sentence similarity and ranking algorithm outperforms other combinations in all scenarios. The best performing combination also varies across datasets. To solve this, we propose using an ensemble of all combinations instead of using any combination in particular.

Table 4.4: Effect of Text representation scheme on ROUGE-Score

		Greedy	Graph	Centroid
Cosine	TF	0.34189	0.35377	0.37271$^{*\diamond}$
	W2V	0.32246	0.31233	0.33254*
	TS	0.35923$^{\diamond}$	0.36245$^{*\diamond}$	0.35634
	LSA	0.31128	0.32528	0.31456
Word Overlap	TF	0.35949*	0.35152	0.35869
	TS	0.36855$^{*\diamond}$	0.35952$\diamond$	0.36222$\diamond$

* Indicates the best performing text representation scheme for a given sentence similarity measure and ranking algorithm

$^{\diamond}$ Indicates the best performing ranking algorithm for a given sentence similarity measure and text representation scheme

There are three possible ways of creating an ensemble system: by developing a new algorithm that takes inputs from various components and combines them into a new system, by combining already generated summaries, or by combining several rank lists of sentences. In the first case, new sentence similarity measures or ranking algorithm can be developed by using the existing ones. [55] use a similar technique by combining five different sentence similarity scores to create a new sentence similarity measure. They then use it with a graph-based approach to generate a summary. Although, it might not always be possible to normalise all the similarity metrics and combine them meaningfully. It would be even more difficult to combine ranking algorithms in this manner. The second method is usually dependent on the availability of training data like in [32] and [64]. We have already discussed the limitation of this method in Related works section.

In this chapter, we experiment with the third approach. Any ensemble system that takes into account the original systems and does not solely depend on the summaries produced by them can be built by combining the rank lists generated by the individual systems. At the same time merely combining all the rank lists do not always guarantee improved performance. We explore the merits of ensembles systematically generated from candidates, rather than blindly combining all the rank lists.

4.3.1 Effect of informed fusion

In this experiment, we demonstrate the individual effect of the three components of a summarisation system: sentence similarity metric, ranking algorithm and text representation scheme. As in the previous experiment, we first generate various combinations by varying either the sentence similarity metric or the ranking algorithm while keeping rest of the setup same. We then try to answer the question: what would be more effective, ensemble using a variation of these similarity metrics or variation of the ranking algorithm, or both. We also conduct a similar experiment where the text representation schemes are varied keeping the sentence ranking algorithm and the similarity metric constant. We experiment with four text representation schemes namely tf-idf, LSI, word2vec and topic signatures.

We create a meta-system by combining the rank lists of sentences from individual systems and creating a single common rank list. The top-k sentences can then be chosen to create the new summary. Three techniques are used for generating the new rank lists:

- Borda Count: Each sentence in the original rank lists are given a score equal to their rank, i.e. sentence ranked first is given a score 1, the one ranked second is given a score 2 and so on. The aggregate score is computed by averaging the score of a sentence in all the rank lists.

- Reciprocal Rank: This is similar to Borda count except that each sentence is assigned a score equal to the inverse of its rank, i.e. first sentence has a score 1, the 2nd sentence has a score 0.5 and so on. It is different from Borda count in the sense that difference between sentences ranked lower is less, and they are penalised almost similarly.

- Weighted Consensus Summarization: This technique, proposed by [80], tries to minimise the weighted Euclidean distance between consensus rank list and the candidate rank lists under the constraint of smoothness of weights. A detailed explanation of this technique has already been discussed in *related work* section.

Table 4.5 shows the results when meta system was constructed by varying either ranking algorithm or similarity metric. The results show that meta-systems created by variation in sentence similarity metric tend to perform better than the individual systems on the DUC as well as Legal datasets. In case of Legal and ACL datasets, we do not use the Greedy algorithm, since the document size for these corpora is too large. In case of greedy algorithm , the time taken to generate summary increases exponentially with the increase in number of sentences. The ensemble created in such a manner always has a higher ROUGE score than the individual systems. On the other hand variation in a sentence, ranking algorithm does not have a significant effect. While the ensemble still performs better than most candidate systems, it does not necessarily outperform the one with the best performance. We used a two-sided sign test with $\alpha = 0.05$ to verify that the difference is statistically significant. There was no significant improvement in case of the ACL corpus. We further compare the results of our metasystem with two state of the art extractive summarization systems *Determinantal Point Processes* proposed in [37] and *Submodular* proposed by [43]. The Determinantal Point Processes (DPP) system is a set of probabilistic models which select important information while maintaining the length of sentences and diversity among selected sentences. [37] provide a probabilistic framework for optimizing opposing objectives (information vs diversity). This makes it possible to train the system using a naive Bayes classifier or other machine learning techniques when some training data is available. The work by [43] poses extractive summarization as a subset selection problem $S \subseteq D$ such that:

$$S \in \underset{X \subseteq D}{\operatorname{argmax}} f(X) \text{ subject to: } \sum_{i \in B} c_i \leq b \tag{4.1}$$

Here c_i is the cost of a sentence that can be defined, for example, as the number of words in the sentence, while b is defined as the total budget of the summary, for example, maximum number of words allowed. This problem is then posed as a submodular optimisation problem. Each constraint is now treated as a submodular 'shell' function. A mixture of such shells is then used to produce the overall submodular function. We compare the results obtained using DPP and Submodular techniques on DUC 2004 datasets to our ensemble system in Table 4.6. DPP and Submodular outperform our ensemble system by a significant margin. However, considering that the aim of our work in not to build

a state of the art summarisation system but is to demonstrate active ensemble of existing systems, it is unfair to compare the results directly with the state of art systems. Instead, we highlight the fact that if these state-of-art systems were to be included in the ensemble, the performance of metasystem would beat all the individual systems. As shown in Table 4.6, the new metasystems that include the state of art systems, far outperform both DPP and Submodular. DPP and Submodular will have same ROUGE scores across all columns, because we have not considered any variation or tweaking in these systems, and instead use them in their original form for the sake of comparison. The Borda2 system in 4.6 refers to an ensemble of the three original candidate systems as well as DPP and Submodular.

Table 4.7 shows the effect of variation in text representation scheme. We also include results obtained by using ROUGE-L besides ROUGE-1. The ROUGE-L results are shown in table 4.8. The variation in text representation schemes does not improve either of the ROUGE scores in most cases when using cosine similarity. The only improvement we note is in case of the graph-based system. On the other hand, when using word overlap, the ensemble across text representation techniques outperforms the individual candidate systems. On the surface, it appears that variation in text representation scheme is not suitable for creating an ensemble technique. However, looking closely at the results, the LSA and word2vec based techniques fare very badly. This is the actual reason behind the poor performance of the ensemble system. The nature of all the ensemble techniques used in this chapter is such that as long as the rank lists agree to an extent, the metasystem will be able to leverage their differences and produce a better ensemble. However, in cases where the candidate systems are very different, the performance of metasystem is not guaranteed. This hypothesis is further confirmed by the leave-one-out test, where we create an ensemble of only a subset of the techniques while leaving one (or two) systems out. The ROUGE-1 scores are shown in Table 4.10. The ensemble of only the systems using Term frequency and Topic signatures outperforms both the individual techniques. However, including Word2vec or LSA based representation hurts the performance. Either these techniques are not efficient text representation schemes for a summarisation task, or a new sentence similarity measure that can make better use of these representations needs to be developed. We report a similar experiment in case of LEgal datasets. Beside tf and

Table 4.5: Effect of sentence similarity metric/ranking algorithm on meta-system

		Greedy	Graph	Centroid	Borda	RR	WCS
DUC 2002	Cosine	0.41785	0.43321	0.44974	0.45109†	0.45133†	0.45734†
	Word Overlap	0.43012	0.42214	0.41806	0.43896†	0.43516	0.44222†
	KL-Div	0.41500	0.41237	0.42234	0.42056	0.42416	0.42796
	Borda	0.44387‡	0.44264‡	0.45213			
	RR	0.44632‡	0.45022‡	0.45813‡			
	WCS	0.45213‡	0.45880‡	0.46239‡			
DUC 2003	Cosine	0.33726	0.34292	0.34382	0.34499	0.34343	0.34714
	Word Overlap	0.32252	0.33566	0.28603	0.33896	0.33916	0.34222†
	KL-Div	0.33892	0.33234	0.32366	0.34056	0.34016	0.34396†
	Borda	0.34987‡	0.36132‡	0.35213‡			
	RR	0.34987‡	0.36222‡	0.35498‡			
	WCS	0.35623‡	0.37246‡	0.36187‡			
DUC 2004	Cosine	0.34189	0.35377	0.37271	0.36490†	0.36042†	0.36764†
	Word Overlap	0.35949	0.35152	0.35869	0.35554	0.34955	0.36222
	KL-Div	0.34160	0.34584	0.35356	0.34580	0.34106	0.35946
	Borda	0.36108	0.36282‡	0.37501			
	RR	0.36089	0.36100‡	0.37328			
	WCS	0.37356‡	0.37659‡	0.37927‡			
Legal	Cosine	-	0.65821	0.64411	0.3	0.65011	0.65222
	Word Overlap	-	0.64321	0.65310	0.64801	0.65400	0.65200
	KL-Div	-	0.63912	0.64900			
	Borda	-	0.65709	0.65200			
	RR	-	0.66001	0.66025			
	WCS	-	0.67232†	0.66986†			
ACL	Cosine	-	0.35412	0.36210	0.36230	0.35920	0.36280
	Word Overlap	-	0.30523	0.31249	0.30405	0.30726	0.31549
	KL-Div	-	0.33250	0.34264	0.34111	0.33910	0.35002
	Borda	-	0.31000	0.32320			
	RR	-	0.31232	0.34003			
	WCS	-	0.32040	0.34213			

† indicates significant difference ($\alpha = 0.05$) between the meta system and best performing candidate system for a given similarity score, for a given year.

‡ indicates significant difference ($\alpha = 0.05$) between the meta system and best performing candidate system for a given ranking algorithm, for a given year.

TS representations, we also use the LCP[45] and YAKE representations[6]. Both these are similar to the TS representations, but differ in the way topic signatures (or *catchphrases*) are computed. The results for LCP based technique were the best, and no significant improvement was achieved by using ensemble techniques. The results are reported in table 4.9. As in the previous case, we do not report the results for Greedy algorithm on Legal dataset.

Table 4.6: Comparison of the metasystem and state of art systems (DUC 2004)

	Greedy	Graph	Cosine
Cosine	0.34189	0.35377	0.37271
Word Overlap	0.35949	0.35152	0.35869
KL-Div	0.34160	0.34584	0.35356
Borda	0.36108	0.36282	0.37501
RR	0.36089	0.36100	0.37328
WCS	0.37356	0.37659	0.37927
DPP	0.39790	0.39790	0.39790
Submodular	0.39180	0.39180	0.39180
Borda2*	0.42315	0.42777	0.42820
RR2*	0.42002	0.42766	0.42710
WCS2*	0.43312†	0.43450†	0.43560†

† indicates significant difference ($\alpha = 0.05$) between the meta system and best performing candidate system for a given similarity score, for a given year.
* Borda2 is an ensemble of the three original candidate systems and the two state of art systems using Borda count. RR2 and WCS2 are also computed similarly

4.4 Discussion

Overall, the results indicate a clear trend: an ensemble of systems which vary in sentence similarity metric, keeping other components constant, tend to perform much better than the individual candidate systems. When designing a new extractive summarisation system, it is a good idea to consider several sentence similarity measures, rather than anyone in particular. On the other hand, going solely by the average ROUGE scores, it is evident that this approach does not always work when using multiple sentence ranking algorithm and the ensemble is not significantly better even if it has slightly higher average ROUGE score in some cases. Although this does not rule out the utility of variations in the ranking algorithm. We show that even in cases for which it did not have a higher ROUGE score, the ensemble system was much more robust compared to the individual candidate systems.

4.4.1 Determining the robustness of candidate systems

Usually, a two-sided t-test is used to check whether the best performing system is significantly better than the rest of the systems. However, this is limited to comparison with

Table 4.7: Effect of Text representation scheme on meta-system for DUC 2004 dataset (ROUGE-1)

		Greedy	Graph	Centroid
Cosine	TF	0.639	0.658	0.644
	LCP	**0.672**	0.679	**0.682**†
	YAKE	0.654	0.666	0.648
	TS	0.613	0.627	0.651
	Borda	0.641	0.632	0.657
	RR	0.642	0.625	0.648
	WCS	**0.672**	**0.681**	0.672
Word Overlap	TF	0.649	0.643	0.653
	LCP	0.681	**0.682**	0.677
	YAKE	0.664	0.659	0.649
	TS	0.653	0.644	0.661
	Borda	0.662	0.658	0.643
	RR	0.671	0.658	0.649
	WCS	0.**683**	0.680	**0.682**†

† indicates significant difference ($\alpha = 0.05$) between the meta system and best performing candidate system for a given text representation, for a given year.
‡ indicates significant difference ($\alpha = 0.05$) between the meta system and best performing candidate system for a given ranking algorithm, for a given year.

the best performing system, and comparison between all systems are not usually reported. We observed that better average ROUGE scores for a system are sometimes biased by higher values for a few documents and do not necessarily ensure robustness. There were a few cases where a system with significantly higher ROUGE scores outperforms the lower ranked system by a considerable margin for some documents sets, but the scene changes for a different document set. We argue that, given two systems with similar ROUGE scores, it is better to have a system that works well for all documents in general, rather than the system which has a very high score for some documents and low for others.

We estimate the consistency of candidate systems by computing two scores, average-rank and G-ROUGE. Average-Rank indicates the average number of systems that the given system outperformed, with a certain minimum margin. For example, when considering three systems, rank each system for individual documents (based on ROUGE recall)

Table 4.8: Effect of Text representation scheme on meta-system for DUC 2004 dataset (ROUGE-L)

		Greedy	Graph	Centroid
Cosine	TF	0.29942	0.30829	0.32578
	W2V	0.27328	0.26555	0.28574
	TS	0.30875	0.31328	0.30000
	LSA	0.26010	0.27432	0.26300
	Borda	0.30805	0.31888‡	0.31444
	RR	0.30623	0.30690	0.31628
	WCS	0.31000	319966‡	0.32124
Word Overlap	TF	0.31093	0.30443	0.31001
	TS	0.31623	0.31004	0.31322
	Borda	0.32200	0.31000	0.31850
	RR	0.32200	0.30890	0.32008‡
	WCS	0.33014‡	0.31724 ‡	0.32498‡

† indicates significant difference ($\alpha = 0.05$) between the meta system and best performing candidate system for a given text representation.
‡ indicates significant difference ($\alpha = 0.05$) between the meta system and best performing candidate system for a given ranking algorithm.

and then compute the average of these individual ranks across all documents. Higher the average-rank, more consistent with a given system. While this approach is not very sophisticated and there might be better ways to estimate consistency of the system, it nevertheless is a good approximation which can provide some insights. We also report G-ROUGE, an alternate computation of ROUGE score, where we compute the score by taking a Geometric mean of ROUGE scores across documents rather than arithmetic mean. The idea is similar to GMAP (Geometric Mean average precision), used in Robust information retrieval track at TREC [79]. Unlike arithmetic mean, the geometric mean is known to be sensitive to variance across the scores. For two groups of numbers having the same arithmetic mean, geometric mean favours the group which has less variance.

Table 4.11 shows the average rank and G-ROUGE scores for each of the candidate systems, and the sentence similarity based ensemble systems on the DUC2004 dataset. Table 4.12 shows these results for the ranking algorithm based ensemble systems. The average ranks were computed by comparing the system in the same columns. For example in Table 4.11 Greedy algorithm with cosine similarity has an average rank of 2.13.

Table 4.9: Effect of Text representation scheme on meta-system for Legal Dataset (ROUGE-1)

		Graph	Centroid
Cosine	TF	0.35377	0.37271
	W2V	0.31233	0.33254
	TS	0.36245	0.35634
	LSA	0.32528	0.31456
	Borda	0.37000‡	0.36928
	RR	0.36050	0.36914
	WCS	0.37000‡	0.37258
Word Overlap	TF	0.35152	0.35869
	TS	0.35952	0.36222
	Borda	0.35952	0.36823‡
	RR	0.35952	0.37002‡
	WCS	0.36555‡	0.37560‡

† indicates significant difference ($\alpha = 0.05$) between the meta system and best performing candidate system for a given text representation, for a given year.
‡ indicates significant difference ($\alpha = 0.05$) between the meta system and best performing candidate system for a given ranking algorithm, for a given year.

Table 4.10: Leave-one-out analysis for DUC 2004 dataset (ROUGE-1)

	Greedy	Graph	Centroid
TF + W2V	0.34000	0.33772	0.36457
TF + TS	0.36842†	0.37004†	0.37980†
TF + LSA	0.32222	0.33147	0.36280
W2V + TS	0.34005	0.34823	0.34254
W2V + LSA	0.31428	0.33500	0.32328
TS + LSA	0.33215	0.34368	0.33343
¬TF	0.32888	0.33496	0.32767
¬W2V	0.34232	0.35888	0.35800
¬TS	0.32111	0.35232	0.36380
¬LSA	0.35521	0.35288	0.36282
All	0.36002	0.37000†	0.37258

† indicates significant difference ($\alpha = 0.05$) between the meta system and best performing candidate system for a given text representation

This means that on an average, the combination outperformed 2.13 out of the other five

Table 4.11: Average rank and G-ROUGE for sentence similarity based ensemble (DUC 2004)

	Greedy		Graph		Centroid	
	Avg-Rank	G-ROUGE	Avg-Rank	G-ROUGE	Avg-Rank	G-ROUGE
Cosine	2.13	0.3381	2.22	0.3491	2.73	0.3602
W.O.	1.85	0.3367	2.12	0.3484	2.50	0.3490
KLD	1.91	0.3351	1.83	0.3426	1.95	0.3410
Borda	3.40	0.3608	3.21	0.3594	3.10	0.3662
RR	3.36	0.3593	3.31	0.3591	3.08	0.3621
WCS	4.20	0.3666	4.69	0.3672	4.35	0.3680

Table 4.12: Average rank and G-ROUGE for ranking algorithm based ensemble (DUC 2004)

	Cosine		W.O.		KLD	
	Avg-Rank	G-ROUGE	Avg-Rank	G-ROUGE	Avg-Rank	G-ROUGE
Greedy	2.33	0.3381	2.00	0.3367	1.93	0.3351
Graph	2.29	0.3491	2.15	0.3484	2.20	0.3426
Centroid	2.85	0.3602	2.18	0.3490	2.08	0.3410
Borda	3.21	0.3625	2.93	0.3540	3.08	0.3500
RR	3.18	0.3613	2.97	0.3542	3.00	0.3492
WCS	3.25	0.3645	3.08	0.3605	3.12	0.3535

combinations of the greedy algorithm. It is evident that the average rank and G-ROUGE scores generally agree. In a few cases where these measures do not agree, the difference in performance of the corresponding systems is very low. For example, the Average rank for *Greedy+Word* overlap combination is lower than that of *Greedy+KLD* combination, in 4.11. However, the G-ROUGE score for *Greedsy+Word Overlap* is higher. However, the difference in average rank for both combinations, as well as that in G-ROUGE is minimal. This is true for all the cases where Average rank and G-ROUGE do not agree. In general, we can say that an ensemble system is always more consistent compared to the candidate systems, even for cases where their ROUGE scores were comparable or slightly lower than the candidate systems. The increase, of course, depends on how well the ensemble performed and is higher in the case where ROUGE scores were significantly better.

4.4.2 Qualitative analysis of summaries

Given that the systems considered in this chapter are purely extractive, evaluating the quality of summaries boils down to evaluating the coverage of the generated summaries. Grammatical inconsistencies are ruled out, because we are not modifying the existing sentences in any manner, nor are we generating any new sentences. We report some sample summaries in Appendix B. We observed a difference in average sentence lengths of the top-ranked sentences, across different ranking algorithms. Greedy algorithm usually pulls up longer sentences, and the 100-word summary is limited to two or three sentences. Compared to that graph-based techniques, includes several shorter sentences high up in the rank list. This works very well when there are concise, informative sentences in the original document. However, in several cases, these short sentences are meaningless and can be dropped without any significant information loss. Co-relating average lengths of sentences in a document to the performance of various systems might be worth exploring. For a given document, rank lists from various systems can then be weighted, depending on which system is likely to perform better.

There seems to be a definite trend in the quality of ensemble summaries. The ROUGE scores are higher when the candidate summaries share at least some information. In such cases, the ensemble retains those common sentences, while pulling up other relevant ones. This is evident from the results of document *d30047* shown in Appendix A. But in cases where the candidate summaries had minimal overlap, quality of ensemble summary is low. This is true not only for systems having high variation in information but also for systems having high variation in ROUGE scores. If one of the candidate systems performs very poor compared to the others, the ensemble is usually unable to outperform the best candidate system. As more and more candidate systems perform worse, they, in turn, pull down the performance of the ensemble. This seems to be the case in the experiment where text representation scheme is varied. The results are adversely affected by the poor performance of word2vec and LSA based techniques. The fact that topic signature representation worked very well in both cases (cosine similarity and word overlap), but word2vec and LSA did not, highlights the possible problem. Similarity scores are well defined in case of topic signature representations as this representation is similar to tf-idf.

However, in case of word2vec to define cosine similarity we first need to define a sentence vector. We do that by taking a mean of the word vectors which might not be an appropriate way to represent a sentence. It is similar in case of LSA. It seems that to make these representations useful in our case, we either need to have a better way to aggregate word vectors to represent sentences or we need a new sentence similarity metric that works well with the existing representations. This might be a direction worth exploring in the future.

To conclude, in this chapter we primarily highlight the fact that when proposing a new extractive summarisation technique, the focus should be more on finding a better sentence similarity measure, while the choice of sentence ranking algorithm does not affect the performance much. In general, whenever proposing an extractive summarisation technique, it would be useful to choose an ensemble of more than one sentence similarity metrics. We demonstrate a new approach of using simple variations of existing text summarisation techniques, to create several candidate systems. Under specific constraints, an ensemble of these candidate techniques, built using simple rank aggregation, has the potential to provide a significantly better and consistent performance compared to the individual systems. By varying various components of text summarisation system, we demonstrate that multiple ranking algorithms and sentence similarity metrics lead to a better and more robust meta-system. While the variation in text representation does not always help, but that is mainly due to the inability to combine word level representations to sentence level representations.

CHAPTER 5

Leveraging content similarity in summaries for generating better ensembles

In chapter 4 we described the technique to effectively aggregate rank lists by variation in sentence similarity, text representation and ranking algorithms. This was part of a larger family of *Consensus-based summarisation* systems, that *democratically* select common content from several candidate systems by taking into account the individual rankings of candidates. In this chapter, we highlight the significant limitations of consensus based systems that rely only on sentence ranking and not on the actual content of the candidate summaries[47]. Their inability to take into account relative performance of individual systems and overlooking content of candidate summaries in favour of the sentence rankings limits their performance in several cases. We suggest an alternate approach that can potentially overcome these limitations. We show how, in the absence of gold standard summaries, the candidates can act as pseudo-relevant summaries to estimate the performance of individual systems. We then use this information to generate a better aggregate. Experiments show that the proposed system outperforms existing consensus-based techniques by a large margin.

The major contributions in this chapter are the following:

- Framework for estimating the performance of a summarization system for a given document cluster, in absence of gold-standard summaries
- Novel techniques for estimating local (for a particular document) and global (across all documents) performance of a summarization system

- Aggregation framework for generating a meta-summary from several candidates by weighting their content based on the performance estimates mentioned above

5.1 Limitations of consensus based aggregation

As the name suggests, the consensus-based methods try to generate a meta-summary that is equally acceptable to all candidate systems. The sentences that are broadly accepted by several systems tend to be ranked higher rather than those championed by only some. Such methods work well, under the assumption that all candidate systems are equally good. However, this is not always the case. Not only that, the performance of summarisation systems, like any other system, varies across the documents. A system that is very good on an average can still perform very poorly on some documents. To give an example, we show a simple comparison of two extractive summarisation systems from those used by [31] in their experiments. We pick two extreme systems, in terms of performance, from the those reported in their work. We compared the FreqSum system[58], which has the weakest performance, to the DPP system[37] which performed the best amongst those compared. On DUC 2004 dataset, FreqSum performed better on more than 10% of the document clusters. There are documents for which a system that is overall very weak outperforms the one that, on an average, has good performance. The ensemble strategy proposed by us in the previous chapter does make the system more robust, but it still does not explicitly take into account the variance in performance for a given document. Neither do the other two ensemble techniques. Borda based rank aggregation does not differentiate between candidates. While the WCS system[80] does assign a different weight to each candidate, it imposes a constraint on the smoothness of weights, which ensures their uniformity to the extent possible. Failure to take into account variance in performance of candidate systems across documents is a major limitation of consensus-based methods. A few poor systems can severely limit the overall performance of the ensemble.

Another major limitation of rank aggregation based techniques is their inability to take into account the content of each candidate systems. In a multi-document summarisation setup, especially in case of newswire, it is quite common to have duplicate or near-

duplicate content getting repeated across multiple documents. To put things in a perspective, DUC 2003 dataset has, on an average, 34 sentences per cluster which have an exact match within the documents. More than 50 sentences have an 80% match with another sentence. Most existing summarisation techniques do not handle this redundancy explicitly, neither do most existing aggregation techniques. This has a huge impact on a class of ensemble techniques which rely only on rank aggregation without taking into account the actual content of individual summaries. Simply comparing rank lists of the sentence does injustice, in cases where different systems selected different sentences with very similar information. For instance, consider the example shown below where S_1 and S_2 are individual sentence rankings, and S_A is the aggregate ranking. This would be fine if each sentence is different and equally important. But consider a case where $sim(s_1, s_4) = 1$. The fact that s_1 and s_5 are repeated across documents makes them more important. However, the rank aggregation techniques fail to take into account their actual content and treat these separately, which results in lowering of their aggregate scores. S_A^* indicates the ideal aggregate.

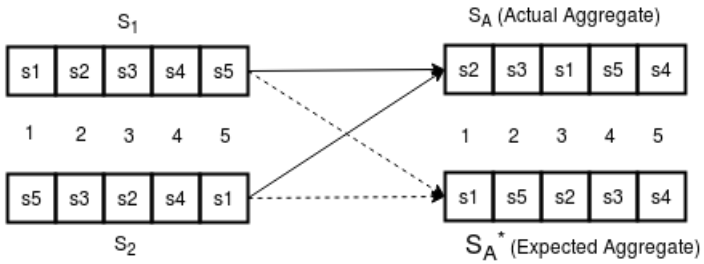

Figure 5.1: Issue with rank aggregation

Instead, in the proposed approach we take into account content of the summaries being aggregated to assign weights to candidate systems.

Another limitation that is specific to the WCS method is the constraint of minimising the distance between entire rank lists. Instead of the top-k sentences which form the summary, WCS tries to optimise the entire ranked list, which is unnecessary. As long as candidate systems agree in the top-k sentences, which are to be considered for the summary, any additional constraint on lower ranked sentences can adversely affect the performance. Moreover, it uses L_1 norm for computing similarity (or distance) between candidate rankings, which can be a sub-optimal choice when compared to traditional met-

rics like *Kendall's Tau*.

In this chapter, we suggest two approaches that take into account content of the candidate summaries and use the similarity between them to estimate the reliability of each candidate for a given document.

5.2 Proposed approach for content based aggregation

Like the aggregation approaches mentioned in the previous chapter, the idea is to use a weighted combination of individual sentence rankings to generate an aggregate ranking. The problem boils down to finding the best combination of weights that maximises the ROUGE score. In the proposed approach we define a new method for assigning weights to different candidate systems. We call this approach *Content based Weighted Consensus Summarization (C-WCS)*. Ideally, a better performing system should contribute more to the aggregate summary compared to a system with lower ROUGE scores. Of course in a practical setup, where the benchmark summaries are not available apriori, it is impossible to know which system will perform better. Theoretically, it is possible to train a system that can predict this information, by looking at the input document. However, in practice, the utility of such a system would be limited by the amount of training data available. Instead of this approach, we propose a completely unsupervised method, which uses the candidate summaries themselves as *pseudo relevant summaries*.

We present a hypothesis that for a summarisation task in general, the *relevant* content in a document cluster is much lower compared to *non-informative* content. Under this assumption, two excellent or *informative* summaries would have a higher overlap in content, compared to two poor summaries. Only because the good summaries will have lesser content to choose from, so they are bound to end up with higher overlap. Based on this we argue that the probability of a candidate summary, that has higher overlap with peers, having good content and in turn, a better ROUGE score is high. The assumption that *good* summaries will have higher overlap amongst themselves, compared to the *weak* summaries, is central to the proposed approach. This condition will not be satisfied, if

two systems that perform poorly, also generate very similar rankings. However, this is not true in general, and we show that there is a very good co-relation between rankings generated using Original ROUGE scores (based on handwritten summaries) and the pseudo ROUGE-scores (based on comparison with peers). While the scores themselves differ very much, the system rankings based on these two scores have a Kendal's Tau of 0.7. This indicates that in the absence of handwritten summaries, a collection of several peer summaries can serve as a useful reference. In the proposed approach the performance of a given candidate is estimated by the amount of content it shares with other candidates. We propose two types of approaches: document level approach (*DocRank*) and sentence level approaches*SentRank and HybridRank*, that uses this estimate to generate a weighted combination of ranked lists.

5.3 Document level aggregation

Consider N candidate systems. For a given candidate summary S_i, each of the remaining $N-1$ candidate systems, $S_j : j\epsilon\{1...N\}, j \neq i$, are considered to be *pseudo-relevant* summaries. We then estimate relative performance of the individual system from the amount of content it shares with these *pseudo relevant summaries*. Weights of a candidate system i is computed as shown in equation 5.1. Sim(S_i,S_j) is defined as ROUGE-1 recall computed considering S_j as the benchmark summary used to evaluate S_i.

$$w_i = \frac{1}{N-1}\sum_{j\neq i} Sim(S_i, S_j) \tag{5.1}$$

The individual rank lists R_i can then be combined using these weights. The underlying assumption in this approach is that the systems performing poorly for a given document are much less in number than the ones performing well. This is not a weak constraint, but we show that this is generally true. In general, a given candidate system tends to perform well on more number of documents compared to the ones on which it performs poorly. Out of the six candidate systems that we experimented with, only one performed below average in more than 30% cases. The number of documents for which more than half candidates performed below average, was only 20%. Given this information, we assert that the number of systems performing well for a given document is generally higher than

Table 5.1: System performance comparison for DUC datasets

System	DUC 2003			DUC 2004		
	R-1	R-2	R-4	R-1	R-2	R-4
LexRank	0.357	0.081	0.009	0.354	0.075	0.009
TexRank	0.353	0.072	0.010	0.356	0.078	0.010
Centroid	0.330	0.067	0.008	0.332	0.059	0.005
FreqSum	0.349	0.080	0.010	0.347	0.082	0.010
TsSum	0.344	0.075	0.008	0.352	0.074	0.009
Greedy-KL	0.339	0.074	0.005	0.342	0.072	0.010
Borda	0.351	0.080	0.0140	0.360	0.0079	0.015
WCS	0.375	0.088	0.0150	0.382	0.093	0.0180
C-WCS	0.390	**0.109**†	0.0198	**0.409**†	**0.110**	**0.0212**
Oracle	**0.394**	0.104	**0.0205**†	0.397	0.107	0.0211
Submodular	0.392	0.102	0.0186	0.400	**0.110**	0.0198
DPP	0.388	0.104	0.0154	0.394	0.105	0.0202

Figures in bold indicate the best performing system
† indicates significant difference with $\alpha = 0.05$

the ones that perform poorly. So subsequently, our hypothesis holds.

5.3.1 Experimental results

As in the previous experiments, we use DUC 2003 and DUC 2004 datasets for evaluating the experiments and report ROUGE-1, ROUGE-2 and ROUGE-4 recall. We continue to experiment with the same candidate systems as used earlier: Lexrank[20], Textrank, Centroid[67], FreqSum[58], TopicSum[41] and Greedy-KL[29]. We use three baseline aggregation techniques against which the proposed method is compared. Besides Borda Count and WCS, we also compare the results with the *choose-best* Oracle technique. In case of the Oracle method, we assume that the performance of each candidate system, in terms of ROUGE score, is known to us apriori. For each document, we directly select the best candidate summary and call it the meta-summary. This is a solid baseline with an average ROUGE-1 recall of 0.394 on the DUC 2003 dataset compared to 0.357 for the best performing LexRank system. We further compare the results with two state of the art extractive summarisation systems Detrimental Point Processes[37] and Submodular[43]. The results are shown in table 5.1 below.

For DUC datasets, in all cases the proposed C-WCS system outperforms other consensus-based techniques, Borda and WCS by a significant margin. It performs at par with the

Table 5.2: System performance comparison for Legal and ACL datasets

System	Legal			ACL		
	R-1	R-2	R-4	R-1	R-2	R-4
LexRank	0.658	0.350	0.155	0.354	0.087	0.020
TexRank	0.643	0.342	0.160	0.305	0.074	0.018
FreqSum	0.629	0.344	0.160	0.331	0.088	0.018
TsSum	0.670	0.355	0.178	0.266	0.055	0.012
Borda	0.635	0.343	0.163	0.298	0.062	0.015
WCS	0.662	0.362	0.175	0.308	0.073	0.018
C-WCS	0.710	0.373	0.186	0.315	0.082	0.016
Oracle	0.700	0.382	0.192	**0.397**†	0.089	0.022
Submodular	0.695	0.382	0.187	0.360	0.087	0.021
Neural	**0.715**	**0.415**†	**0.215**†	0.344	**0.090**	**0.027**†

Figures in bold indicate the best performing system
† indicates significant difference with $\alpha = 0.05$

current state of art Submodular and DPP systems. In several cases, C-WCS even outperformed the Oracle system, which relies on apriori knowledge about which system will perform the best. A two-sided sign test was used to compare the C-WCS system with other systems. † indicates that the best performing system is significantly better than the next best performing system. A contrasting result was obtained for the Legal dataset. In this case, the Neural sentence extraction technique proposed in chapter 3, always outperforms all the ensembles by a huge margin. Clearly in case of Legal corpus, the Neural technique is much more robust. One reason for this can be the fact that the neural sentence extractor incorporates domain knowledge, which none of the other techniques have. For ACL corpus again, Neural sentence extractor performs the best for ROUGE-2 and ROUGE-4, while the submodular[43] technique gives the best result for ROUGE-1. In this case however, the ensemble techniques perform very poorly. The reason is, abstracts of scientific articles are very precise and none of the existing techniques are good enough to generate a decent abstract, since all of them look at coverage of the summary and not the template based extraction, which is implicitly learnt by the neural approach. This also highlights the limitation of our ensemble approach, which will fail if the candidate summaries are too divergent in terms of content. We do not further experiment on the ACL corpus for the remaining techniques in this chapter.

5.4 Sentence level aggregation

In this approach, we propose a new method for jointly estimating the authority of a particular system for a given document and also the importance of each sentence within the summary generated by that system. The ensemble summary is then a function of the authority of each candidate system as well as the relative importance of each sentence in the candidate summaries. We make use of the same hypothesis mentioned in the previous section, that informative or *summary worthy* sentences in a document cluster are much less in number compared to the non-informative ones. We argue that since this content is much less, any substantial overlap between two summaries will likely be due to the *important* content rather than the redundant one.

We use graph-based ranking that takes into account the similarity of a candidate summary with other candidates to generate its local (or document specific) ranking. We also determine the overall global ranking of a system from its ROUGE score on a development dataset. In the same way, *informativeness* of a sentence is linked to its overlap with sentences of other summaries. The HybridRank model proposed here combines these three factors to generate a new aggregate ranking of sentences.

5.4.1 SentRank

This system takes into account similarity of each sentence in a candidate summary, with that of other candidate summaries, and uses it to assign relative importance to the sentence. The argument presented above, favouring the use of similarity as a measure for the reliability of a summary, can be extended at the sentence level. A sentence that is *summary worthy* will share more information with another *summary worthy sentences*.

For i^{th} sentence in the j^{th} candidate summary(s_{ij}), we find the best matching sentence in the remaining candidate summaries. The score of that sentence can then be computed as shown in equation 5.2 below. The score of the sentence is the sum of its similarity with the best matching sentences from remaining candidates. Here j,k are the candidate systems. i

and l are the sentences in candidate J and K respectively.

$$\boldsymbol{R}(s_{ij}) = \sum_{k,k\neq j} \max_{l}(Sim(s_{ij}, s_{lk})) \tag{5.2}$$

Next, each sentence in the candidate summary is ranked according to their score $\boldsymbol{R}$ and top k sentences are selected in the summary. We experimented with n-gram overlap, cosine and KL Divergence for computing the similarity between sentences and empirically select cosine similarity, which is also used in the following systems.

5.4.2 GlobalRank

One limitation of the SentRank approach proposed above is that does not take into account the reliability of candidate systems into account and treats each candidate equally. A sentence that comes from a well-performing candidate system is more likely to be *informative* compared to a sentence from a poor summary. The proposed *GlobalRank* system does exactly that. It builds over the SentRank system by incorporating a candidate's *global reputation* score into the sentence ranking scheme. The new scoring mechanism is shown in equation 5.3 below. $G(k)$ refers to the global reputation of candidate system k.

$$\boldsymbol{R}(s_{ij}) = \sum_{k,k\neq j} G(k) * \max_{l}(Sim(s_{ij}, s_{lk})) \tag{5.3}$$

$G(k)$ is estimated using the average ROUGE-1 recall of each candidate systems as shown in equation 5.4 and 5.5 below. $R1_k$ is the rouge-1 recall of the k^{th} candidate. $R1^{'}$ is the normalised version of $R1$. Here we do not subtract mean, to avoid negative values in scoring, and instead subtract the minimum of $R1^{'}$. Additionally, we scale it using a scaling factor a, which is dependent on the total number of candidate systems. We empirically set a to 0.1. We used the results on the DUC2002 dataset for estimating the ROUGE-1 recall and in turn the GlobalRank of a candidate.

$$G(k) = aR1'(k) \tag{5.4}$$

$$R1_k^{'} = \frac{R1_k}{\sigma(R1_k)} - \min_{k}\left[\frac{R1_k}{\sigma(R1_k)}\right] \tag{5.5}$$

As compared to SentRank, which can be overwhelmed by too many poor performing systems, GlobalRank provides a smoothing effect, by giving more importance to the systems that are known to perform well generally.

5.4.3 LocalRank

One major limitation of the existing aggregation systems, which we highlighted in section 2, is their inability to predict which candidate system will perform better for a given document cluster. Neither of the systems suggested above, *SentRank* and *GlobalRank*, address this problem. The next system, *LocalRank* tries to mitigate this problem. We do not rely on any lexical or corpus specific features, simply because the training data is not sufficient to estimate these features reliably. Instead, we continue on our line of argument, using the similarity between summaries as a measure of reliability. For a give document cluster, we estimate the reliability of a candidate k from the content it shares with other candidates, and also the reliability of those candidates. We first create a graph with the nodes as the candidate summaries and edge as the similarity between nodes. Each candidate starts with the same reputation score or LocalRank (L). The local rank is then updated iteratively using the PageRank algorithm [62]. The Local rank for a given node is estimated as shown in equation 5.6 below:

$$L(k) = \sum_{j} L(j) * Sim(S_j, S_k) \tag{5.6}$$

$L(k)$ indicates local rank of k^{th} candidate, S_k indicate summary generated by the k^{th} candidate. We use cosine similarity as the similarity score. The overall sentence scores are then computed just like in the GlobalRank algorithm.

$$\boldsymbol{R}(s_{ij}) = \sum_{k,k\neq j} L(k) * \max_{l}(Sim(s_{ij}, s_{lk})) \tag{5.7}$$

5.4.4 HybridRank

While LocalRank is useful for estimating how well a given candidate might perform for a given document cluster, it does not make use of the actual system performance.

HybridRank overcomes that limitation. As the name suggests, HybridRank combines strengths of both GlobalRank as well as LocalRank by taking a weighted combination of both. The HybridRank is defined in the equation 5.8 below.

$$H(k) = \alpha L(k) + (1 - \alpha)G(k) \tag{5.8}$$

Here the value of α determines the balance between Local and GlobalRank. A high value of Alpha gives more importance to the estimate of how good a system will perform on a particular cluster while ignoring the overall aggregate performance of the candidate. $\alpha = 0$ leads to the original GlobalRank, without any local information. We empirically set the value of α to 0.3. Once the systems are ranked, the sentence rankings are computed in the same manner as LocalRank or GlobalRank (equation 5.3 and 5.7).

5.4.5 Experimental results

We report the experimental results on the DUC 2003 and DUC 2004, Legal and ACL datasets using the standard ROUGE scores ROUGE-1, ROUGE-2 and ROUGE-4 recall. We use eleven candidate systems which are a mix of several state of the art extractive techniques and other well-known baseline systems. Apart from the systems mentioned in the previous section, we use seven other state-of-art systems. A brief description of these systems is given below.

CLASSY04 Judged best among the submissions at DUC 2004[12], uses a hidden markov model with topic signatures as the features. It links the usefulness of a sentence to that of its neighbouring sentences.

CLASSY11 This method builds over the CLASSY04 technique and uses topic signatures as features while estimating the probability that a bigram will occur in a human-generated summary. It employs non-negative matrix factorisation to select a subset of non-redundant sentences with highest scores.

Submodular [43] treat summarization as a submodular maximization problem. It incrementally computes the *informativeness* of a summary and also provides a confidence score as to how close the approximation is to a globally optimum summary.

DPP Detrimental point processing[37] is the best performing state-of-art system amongst all the candidate systems. DPP scores each sentence individually, while at the same time trying to maintain a global diversity to reduce redundancy in the content selected.

RegSum uses diverse features like parts of speech tags, named entity tags, locations and categories for supervised prediction of word importance[33]. The sentence with the most number of *important* words is then included in the summary.

OCCAMS_V The system by [17], employs LSA to estimate word importance and then use the budgeted maximal coverage and the knapsack problem to generate sentence rankings.

ICSISumm treats summarization as a global linear optimization problem[25], to find globally best summary instead of selecting sentences greedily. The final summary includes most important concepts in the documents.

We use the same ensemble techniques as before, *Borda count* and *Weighted consensus summarization*[80] for comparison. For the GlobalRank system, we used DUC 2002 dataset as a development dataset to estimate the overall performance of candidate systems. We ranked the systems based on ROUGE-1 recall scores for this purpose. The results are shown in table 5.3.

As shown in the table 5.3, for DUC datasets the proposed systems outperform most existing systems on all three ROUGE scores. Both Borda and WCS failed to outperform the best state of art results. Even the simple SentRank algorithm outperforms most candidate systems and achieves a performance at par with the state of art systems in terms of ROUGE-2. While HybridRank achieves the best performance for ROUGE-1 and ROUGE-2 on both DUC 2003 and DUC 2004 datasets, GlobalRank performs best in terms of ROUGE-4 on DUC 2004. We performed a two-sided sign test for determining whether the results were significantly different. The results clearly show that a rank aggregation technique that takes into account content of the summaries achieve a much higher ROUGE score vis-a-vis the systems that use only the ranked lists of sentences. On the contrary, both WCS and Borda fail ot outperform Neural model, as seen in table 5.4. The SentRank and LocalRank show and improvement over baselines but do not outperform the

Table 5.3: Results of sentence level aggregation on DUC datasets

	DUC2003			DUC2004		
System	R-1	R-2	R-4	R-1	R-2	R-3
LexRank	0.3572	0.0742	0.0079	0.3595	0.0747	0.0082
FreqSum	0.3542	0.0815	0.0101	0.3530	0.0811	0.0099
TsSum	0.3589	0.0863	0.0103	0.3588	0.0815	0.0103
Greedy-KL	0.3692	0.0880	0.0129	0.3780	0.0853	0.0126
CLASSY04	0.3744	0.0902	0.0148	0.3762	0.0895	0.0150
CLASSY11	0.3730	0.0925	0.0142	0.3722	0.0920	0.0148
Submodular	0.3888	0.0930	0.0141	0.3918	0.0935	0.0139
DPP	0.3992	0.0958	0.0159	0.3979	0.0962	0.0157
RegSum	0.3840	0.0980	0.0165	0.3857	0.0975	0.0160
OCCAMS_V	0.3852	0.0976	0.0142	0.3850	0.0976	0.0133
ICSISumm	0.3855	0.0977	0.0185	0.3840	0.0978	0.0173
Borda Count	0.3700	0.0738	0.0115	0.3772	0.0734	0.0110
WCS	0.3815	0.0907	0.0120	0.3800	0.0923	0.0125
SentRank	0.3880	0.1010	0.0163	0.3870	0.1008	0.0159
GlobalRank	0.3562	0.1045	0.0185	0.3955	0.1039	**0.0191**†
LocalRank	0.3992	0.1058	0.0192	0.3998	0.1050	0.0187
HybridRank	**0.4082**†	**0.1102**†	**0.0195**†	**0.4127**†	**0.1098**†	0.0180

Figures in bold indicate the best performing system
† indicates significant difference with $\alpha = 0.05$

Table 5.4: Results of sentence level aggregation on Legal dataset

	Legal		
System	R-1	R-2	R-4
LexRank	0.658	0.350	0.155
FreqSum	0.629	0.344	0.160
TsSum	0.670	0.355	0.178
Submodular	0.695	0.382	0.187
Neural	0.715	0.415	**0.215**
Borda Count*	0.652	0.358	0.162
WCS*	0.675	0.372	0.179
SentRank	0.682	0.372	0.183
GlobalRank	0.718	0.413	0.209
LocalRank	0.708	0.400	0.192
HybridRank	**0.725**†	**0.417**	0.213

* Ensemble does not include Neural approach

Neural approach. However, GlobalRank always gives a performance similar to the Neural approach. Given the large difference in performance between Neural approach and other candidate systems, this is intuitive, since alway Neural performs better, which gives it a higher weight in global ranking. HybridRank outperforms Neural approach for Rouge-1 but does not have a significant difference for Rouge-2 and Rouge-4.

5.5 Conclusion

To conclude, in this chapter we describe a novel method for consensus-based summarisation, that takes into account content of the existing summaries, rather than the sentence rankings. For a given candidate summary we treat other peer summaries as pseudo-relevant model summaries and use them to estimate the performance of that candidate. Each candidate is weighted based on their expected performance when generating the meta-ranking. We proposed document and sentence level techniques. In the document level technique (DocRank) we use the overall summaries generated by the candidates to measure overlap, as compared to sentence level overlap in the latter. For sentence-level techniques, we further define three systems, *SentRank*, *LocalRank* and *GlobalRank* which take into account *informativeness* of individual sentences, the performance of candidates on a given document cluster, and overall performance of candidates on a held out development set, respectively. We use content overlap between summaries generated from several systems to estimate the relative importance of each system in case of LocalRank and SentRank. We combine the information from all these three systems to generate the final hybrid ranking (HybridRank) system. Summaries generated from such an aggregate system, both document level and sentence level, outperforms all the baseline and state of the art systems as well as the baseline aggregation techniques by a significant margin, in case of the DUC dataset. However, most ensemble techniques fail to outperform the Neural sentence extraction approach, which implicitly acquires domain knowledge and hence generates much better summaries. The limitation of these approaches is highlighted in the fact that they perform poorly on the ACL corpus, where the summaries are more divergent and the shared content across them is much less.

CHAPTER 6

Neural model for sentence compression

The techniques we have discussed so far focus on improving the *informativeness* of extractive summaries. The neural sentence extraction model discussed in chapter 3, as well as the ensemble, approaches in chapter 4 and 5, all solely focus on choosing a subset of sentences which gives the best ROUGE scores. However, like with any extractive techniques in general, these approaches have a limitation when generating a summary of fixed size. In the absence of reliable generative techniques, which can generate new concise sentences the next logical step is to eliminate redundant or less informative content from the extracted sentences. The two possible ways to achieve this is sentence compression and sentence simplification. While sentence compression solely deals with removing redundant information, sentence simplification usually looks into replacing a difficult phrase or word with a simpler alternative. In case of legal documents, usually replacing long legal phrases with more commonly used phrases also leads to sentence compression. This improves the precision of fixed length summaries.

In this chapter, we begin by presenting a new approach for sentence compression for legal documents. We demonstrate how a phrase-based statistical machine translation system can be modified to generate meaningful sentence compressions. We compare this approach to an LSTM based sentence compression technique proposed in [21]. Next, we show how this problem can be modelled as a sequence to sequence mapping problem, thus not limiting to just deleting words, but also having a possibility of introducing new words in the target sentence. The strength of this approach is that it is entirely data-driven, which means it does not depend on any linguistic resource like most other sentence compression or fusion techniques. At the same time, unlike other deep learning based approaches like

[8] or [9], the proposed approach requires much less training data. The apparent constraint of this technique is that it will work only in cases where the compressed sentences can be formed by merely extracting phrases and introducing some new words, but without much re-writing. This reflects in the results as well, where this approach works very well on the Legal corpus but is of limited use on the ACL corpus. Below we explain two basic models based on neural networks. While the first model learns to delete words from sentences to achieve compression while the other model is capable of adding/replacing words from existing sentence.

Overall the major contributions from this chapter are:

- Sentence compression using statistical machine translation techniques
- Attention based neural model for sentence compression

6.1 Sentence compression by deletion

We use the LSTM based sentence compression model proposed in [21] as a baseline. The overall idea in this model is to treat sentence compression as a sequence labelling problem. Each word in the input sentence is assigned a label '0' or '1' indicating, whether or not the word should be retained in the compressed sentence. The original work uses a parallel corpus of two million sentence-compression instances from a news corpus. Relatively, our corpus is much smaller $\sim$250K sentence-compression pairs for legal corpus and $\sim$150K pairs for the ACL corpus. However, unlike newswire or ACL corpus, the sparsity in the Legal corpus is much lower. The structure of legal sentences does not change much across the documents which makes it possible for us, to a certain extent, to use the sentence compression technique mentioned in [21]. This approach relies on a simple sentence encoder, which creates a sentence embedding using an LSTM based encoder. This is followed by a softmax classifier which sequentially assigns one of three labels to each word using this sentence embedding along with label information of the previous word. In line with the original work, we did not use any PoS or syntactic features. The original work shows that such features only marginally improve the results. The general architecture is shown in

figure 6.1 below.

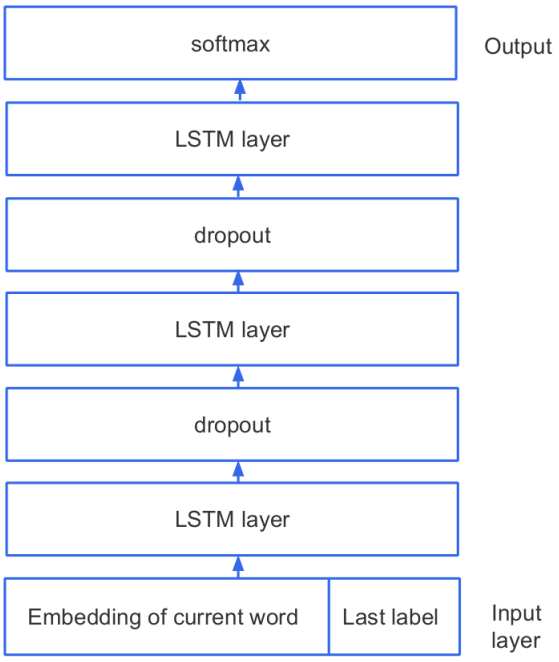

Figure 6.1: LSTM based word deletion model

The sentence encoder uses three stacked LSTM layers for generating the sentence embeddings. There are two significant differences in this sentence encoder when compared to the one proposed in chapter 3. First, this model does not use attention and relies only on LSTM layers to encode the sentence. Second, it uses a two-pass system, where the sentence is passed once to the encoder, generating a sentence embedding. At the end of the first pass, the labelling commences where the entire sentence is once again passed sequentially, and the softmax layer predicts one of three labels for each word. In the second pass, the predicted label of the previous word is concatenated to the current word embedding before passing it to the encoder. We used the same embedding size of 256, as used in the original work. In this case, the input vector will be of 259 dimensions including the one hot representation of the label of the previous word. As opposed to the original work, which uses pre-trained skip-gram model[52] we trained a skip-gram model separately for the Legal and ACL corpora. We limit the maximum sentence size to 200 for the legal corpus, as opposed to 120 in the original work. This is only because sentences in legal documents are much longer on an average, as compared to those in newswire articles. For

the ACL corpus, we use the original limit of 120 words per sentence. We retain all other parameters as in the original work.

6.2 Sentence compression using Sequence to Sequence model

The sequence of sequence architecture was first proposed by [77] in 2014. It has since been used in a variety of applications which require generating sequential data, like speech recognition, machine translation or as in our case sentence compression. The overall architecture of a sequence to sequence model proposed by [77] is shown in figure 6.2 below. Broadly speaking, such a model consists of three modules, an encoder, a decoder and an optional attention module. The encoder is responsible for creating a fixed dimensional encoded representation of the input sequence. The decoder then uses this fixed length representation to generate the output sequence. One significant advantage of this architecture over the phrase based sentence compression discussed in the previous section is the ability to take into account long-range dependencies in an input sentence due to the lstm module. The attention module, on the other hand, defines the local context and learns to 'focus' on certain parts of input sentences, while generating the compressed sentence. We briefly explain the three modules below. This model is similar to the one used in [56], except that we do not use pointer generators. The original work shows that in lack of sufficient training data, using pointer generators decreases the performance. We replace the additional keyword encoder used in their work with a context encoder, which is described below. The additional context encoder represents the metadata in case of legal documents. We report results with and without the context encoder for the legal corpus. We do not use the context encoder for ACL corpus.

6.2.1 Sentence Encoder

Sentence encoder is responsible to sequentially read the word representations and generate intermediate sentence representations for each state. These intermediate representations capture *abstract* meaning of the sentence up to that point. Te word representations, which are used as input to the sentence encoder, can be either one hot encoded vectors of the size

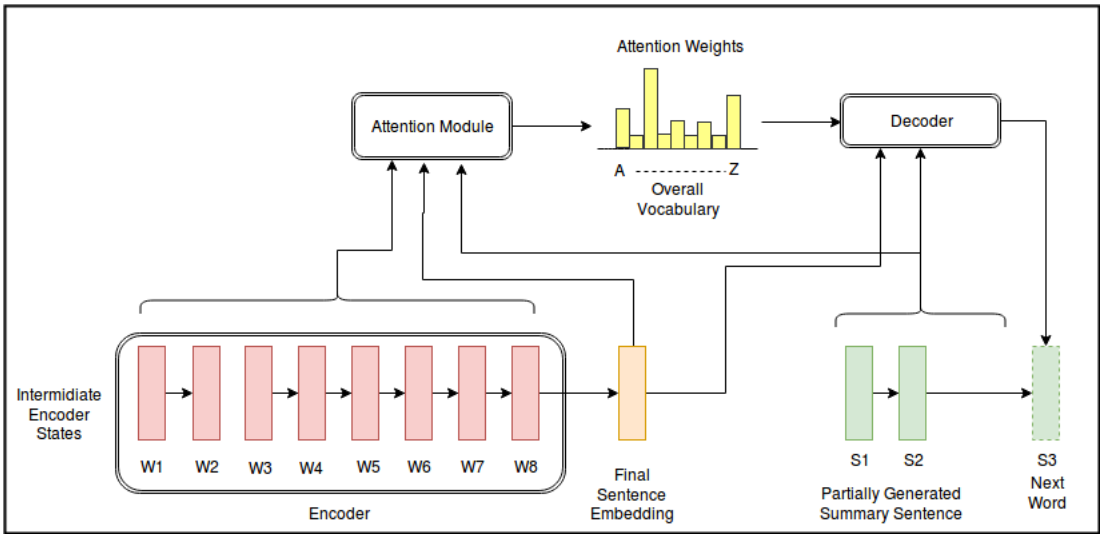

Figure 6.2: Sentence compression model

same as the vocabulary size or word embeddings. In our case, we use word embeddings as the input. As discussed previously we train two in-house skip-gram models on the Legal and ACL datasets to generate these embeddings. We use a two-layer LSTM to create the sentence embeddings. Intermediate sentence representations after each time stamp are retained.

6.2.2 Context Encoder

In general, a sequence to sequence model does not require a document level context embedding. Nonetheless, in our case, such a context vector is quite useful. In case of legal documents, we have some key terms related to the judgement which can be very useful in generating good summaries. However, such case-specific terms are also less frequent and hence prone to being ignored by the neural network which uses a limited vocabulary. We encode this information in the form a context vector of a fixed length of 150. This length was chosen empirically. The first ten dimensions represent the first and last names of the judges delivering the judgement. In case the panel has less than five judges, some of these entries will be zero. Next four dimensions encode the name of the chief appellant for the trial and principal respondent respectively. In case of an organisation being the appellant or defendant, the entire name is used as a single entry in the vocabulary. In this case, there is no last name, and the corresponding entry is zero. The context vector also includes the names of lawyers for both sides, in the same manner as above. Rest of the

dimensions in the context vector represent the case numbers of the cases that were cited in this judgement. We do not use the context embedding in case of ACL corpus.

6.2.3 Decoder

The decoder module generates output sentence one word at a time. It used the final encoder output, the weighted average of the intermediate encoder states (using attention weights) as well as the previous word predicted by the decoder. We use the standard beam search algorithm with a beam size of three for the decoding process.

6.2.4 Attention module

The attention module is primarily responsible for defining the local context within a sentence. It is generally used as a bridge between the encoder and decoder. Attention module, as we use it, represents a weighted average of the intermediate encoder states. At a given step in the decoding process, the attention module assigns weights to each of the input steps. In a way, it decides what part of the input sentence is the most important when generating the next output word. The input to an attention module is the final input embedding, intermediate input embedding states and the last decoder output step. Additionally, we provide the list of entities essential to the case, to the attention module in the form of the context embedding defined above.

6.3 Exploiting SMT techniques for sentence compression

We use the traditional statistical machine translation technique, specifically phrase-based translation as another baseline for our experiments. Here we treat sentence compression as a machine translation problem. The original sentence represents *Document language* and the compressed sentence represents the *Summary language*. In most cases, a word or phrase in the source language will be aligned to at least one word in the target language, and it is uncommon for a source word to not align with any target words (or will align with the symbolic DEL token in target vocabulary). As a result, the phrase based translation systems restrict the deletion operation. This is enforced by putting a penalty on the alignment between a source word and the DEL token in the target. Such phrase level

alignments will be ranked much lower. However, in our case, there is a high probability that a word needs to be dropped and hence will not align with any target words. Therefore in our case, we relax the penalty and allow more words to be aligned to the DEL token. We use the open source Moses toolkit for this experiment.

6.4 Results for sentence compression

We test the attention based sentence compression module on both Legal and ACL datasets. The evaluation strategy we use for sentence compression is a bit different from that used for evaluating the summaries. Although there is a scope of introducing new words in a sentence, our model inherently learns to retain the words from input sentences while adding only a few new terms. This makes ROUGE susceptible to inaccurately high performance. For example, returning the sentence as it is without compression will achieve a recall close to 1. In contrast, retaining only selected informative terms will still have precision close to 1 even if the sentence is grammatically incorrect. To be fair, we use the accuracy of compression system as an additional evaluation measure. This reflects the total number of compressions that were exactly same as the expected target sentence. We also report the F-1 score in terms of unigram overlap between the expected and actual compressions. The work in [21] follows a similar strategy for evaluation. We compare the three approaches mentioned in the previous section. Additionally, we use the context encoder(seq2seq+) in case of the Legal corpus. We also report the compression ratio for all approaches. As evident from table 6.1, the sequence to sequence model outperforms both the SMT model as well as the deletion based model. The results were encouraging for the Legal dataset but the model seems to be of limited use on the ACL corpus.

	Legal			ACL		
	Acc.	F-1	C.R.	Acc.	F-1	C.R.
SMT	0.17	0.73	22	0.06	0.57	10
Del	0.16	0.70	29	0.08	0.55	21
seq2seq	0.21	0.85	21	0.11	0.48	17
seq2seq+	0.25	0.87	22	-	-	-

Table 6.1: Results for sentence compression

We observe that in case of ACL data, the model only learns to delete some phrases

or does not change anything, returning the sentence as it is. Such transformation constitutes a rather small part of the valid compressions in case of ACL data. This is intuitive since, in scientific abstracts, a sentence can contain information from multiple sentences in the document. As opposed to this the sentences in legal documents and summaries have almost a one to one correspondence. In case of the sentence extraction task, certain key phrases we sufficient to determine the *summary-worthiness* of a sentence. In contrast, the ACL data has much more sparsity when it comes to the sentence compression task. This is not the case for legal documents, where although keyphrases do play an important role, the entire vocabulary is constrained, thus reducing the issue of sparsity in training data. Interestingly, using a simple context encoder significantly improved the scores for the sequence to sequence models. Later in this chapter, we present an analysis of the quality of sentence compressions generated. However, in the next section, we discuss the progress so far with the sequence to sequence model and the current state of art techniques. This is not meant to criticise in any way the existing works but is instead intended to serve as an honest evaluation of what has already been achieved and what needs to be done further.

6.5 Limitations of sentence compression techniques

The work by [77] presented a new model for machine translation. The proposed sequence to sequence model first generates an abstract representation of the input text and then sequentially generates output text from this abstract representation, which in their case is the translated sentence. However, the model can handle, at least theoretically, not only translations but any transformation of texts in general. Another significant advantage of this approach is that it is entirely data-driven, and does not depend on any language specific resource. In a way, this work proved to be a turning point for research in several subdomains of Natural language processing including text compression and text generation, which until then were heavily dependent on linguistic resources. It is a different issue that it introduces another type of dependency, that on large volumes of data, but we discuss that later in the section. Although we are far from achieving the goal of successfully generating text from a set of concepts like humans do, the research in this area has progressed a lot since the sequence to sequence models were first proposed in [77]. We

begin by discussing some of the most prominent works in the direction of text compression while trying to present a diverse overview. The initial attempts were limited to achieving sentence compression by deleting redundant words. The model proposed by [21], which although uses the sequence to sequence learning architecture, is in fact, a sequence labelling problem. They first generate a sentence level encoding by using stacked LSTMs and then use this *abstract representation*, to sequentially predict whether or nor a word should be included in the output sentence. They evaluate the model on a set of 250K sentence compression pairs. The dataset was limited to newswire articles, which is generally the case with most other works as well. They were able to successfully reproduce 30% gold-compressions, as compared to the previous best result of 20%. The only limitation of this approach is that it works best in scenarios where just word deletion is sufficient, like in the case of generating headlines of news articles. We use this model as a baseline in our experiments. The proposed Legal dataset has several instances where deletion of a phrase is sufficient for generating a meaningful sentence compression.

For past few years, the progress in sentence extraction or compression has closely followed that in Neural Machine translation. This helps in quickly porting a new architecture to a different application in this case from NMT to summarisation, but at the same time introduces a handicap in the sense that several issues specific to the new problem are largely ignored. The work proposed in [68] and followed up in [9] was inspired by the attention-based model for NMT introduced in [2]. They proposed a model for sentence level abstract generation, which takes as input the first sentence of a news article and produces the headline of the article. The model uses an attention module, which learns to focus on certain parts of the input sentence when generating a particular output token. This is useful not only in improving the quality of compression, but also provides a way to visualise the reason behind a specific output token. Unlike [77] this work does not restrict itself to word deletion and is in a way, purely generative technique. Although they do restrict the target vocabulary to the vocabulary of the original article. It is unclear how much effect the attention module had on the final results. Both these works[68, 9], used a dataset of 4 million sentence compression pairs, which is not publicly available. It is not entirely clear if this performance will be severely affected by a lower amount of training

data. The largest corpus publicly available is 300K sentence compression pairs, which is less than 10% of this dataset. They report Rouge-1 and Rouge-2 scores as the evaluation measure, which is a major limitation. Later in this chapter, we show that we can achieve very high Rouge score but the low quality of sentences. The authors highlight that the generated headlines are often grammatically incorrect. Unfortunately, there is no mention of the number of gold standard headlines that were successfully regenerated. This is usually the case with works reporting sentence compressions or abstract generation [8, 7, 82, 56]. Although in this work we discuss only a handful of highly influential works, this limitation is, in general, true for most works.

Another work that uses attention based sequence to sequence models is [56]. The work introduces copying -mechanism in sentence compression tasks. Copying-mechanism is a way to handle rare words and OOVs. Instead of using the UNK token while predicting a rare word in output, the network learns to copy the word from its original location, hence enabling the use of rare terms in the output sentence. They also introduce a method of capturing keywords using other features besides word embeddings. This makes it possible to tune the importance of different words, by using tf-idf or other tag-based features in addition to the word-embeddings.

A major takeaway from this work is the fact that the same model when trained on a smaller public dataset of 300K sentence-compression pairs, still generates a ROUGE score comparable to that of a system that was trained on a much larger corpus (4 Million pairs). However, in this case, the simple attention model, without the use of copying-mechanism or additional features performs the best. This raises an important question, how effective several state-or-art techniques are when used with a smaller, more realistic training data. Alternatively, should the focus instead be on developing a simpler model with fewer parameters to be trained? For example, the proposed work introduces two exciting concepts of keyword-based features and copying mechanism, but there is no argument about whether this is required at all, and how much the quality (and not ROUGE scores) of the generated summaries will depend on whether or not these mechanisms are used.

The work by [56] was perhaps the first attempt at generating an entire summary, instead of sentence-level compressions. The work by [8] adopts a similar approach, where

the authors create abstracts of news articles by extracting words and sentences. They propose this as a two-step approach, with sentence extraction as the first step followed by word level extraction. Results are reported for both a sentence-level summary as well as a word-level summary. The authors first use convolutional networks for encoding sentence level information and then use a recurrent network to generate document level embedding. The overall architecture is similar to the generic sequence to sequence model, with the significant exception being the use of CNN as sentence encoder. This makes it comparable to [56]. The standard Rouge scores are reported as automatic evaluation measures. Although the neural sentence extractor performs the best, the simple LEAD based baseline has comparable performance. In fact, the Lead-based baseline outperforms the word-level summaries. They also provide manual judgements from humans, where the sentence based extraction is the best performing system with an average ranking of 2.74, closely followed by ILP based technique[10] with an average ranking of 2.77, with the lower ranking being better. The summaries formed by word extraction were poorly rated in terms of ROUGE scores as well as by human evaluators. The work in [83] proposed a two step sentence encoder. The first stage is a general RNN which gives a sentence encoding. The nest stage is a selective gate network which filters out unimportant sentences. The decoder finally generates the compresses sentence. Again an improvement in ROUGE is reported over other techniques but no qualitative analysis is provided.

The systems discussed in this section as a representative sample of the plethora of attempts that have been made in the past few years, and continue to be made. It is clear that the sentence extraction techniques are becoming more reliable, and so are the deletion based techniques. The results in some cases may be comparable to the existing unsupervised methods and simple baselines as shown in [8]. However, in case of domain-specific summarisation, the deep learning based techniques combined with pseudo labelled data will have an edge as shown in chapter 3. At this point, it is not clear how or if the proposed improvements over the basic sequence to sequence model are affecting the performance of sentence generation systems in general. The success of sentence generation is also dependent on large volumes of training data, only a fraction of which is publicly available. The lack of an evaluation measure that looks beyond comparing n-gram overlaps makes the problem worse. In case of deletion based compression, ability to re-generate gold-

compressions is a good indicator of the performance. Such a comparison is not possible for generative techniques. The evaluation in such cases is limited to getting a few samples evaluated by the human annotators which can be affected by several factors like, bias in human annotation, sampling bias (choosing 200 samples from an evaluation set of 10,000 may not be very representative. At this point, it is more important to define a better evaluation measure and look the problem from a more human perspective rather than merely comparing Rouge scores and accuracies. Another important direction is to narrow down the focus to certain domains instead of trying to design a one-size fits all system. Incorporating domain knowledge, and limiting the study to a narrow category of documents can drastically improve the results. In our opinion, these areas of developing a new evaluation technique and focusing on solving specific problems using abstractive summarization deserves more focus than continuing to build new models for headline generation.

Next we briefly present the results on an end to end abstractive system, which uses all the individual modules proposed so far.

6.6 Overall System

The results reported so far only analyse the quality of sentence compression. However, the overall abstract depends equally on sentence extraction and sentence compression. Here, we present the results achieved using an end to end architecture for generating abstracts. The complete pipeline is shown in figure 6.3. The sentence compression block shown in figure 6.3 corresponds to the system described in 6.2 We use the same preprocessing steps as used previously, i.e. we remove stop words and do not perform stemming. We do remove the non-frequent words before the sentence compression module, as described in the previous section. We then use the techniques mentioned in chapter 3 to generate extractive summaries. Optionally one or more of these extracts are combined using the ensemble techniques mentioned in chapter 4 and 5. The original document is used to create a context vector described in the previous section. Finally, each sentence in the extract, along with the context vector, is sequentially given as input to the sentence compression module. The context vectors are created as mentioned in the last section. The compressed sentences are then reordered in the order in which they appear in the original document,

and constitute the summary.

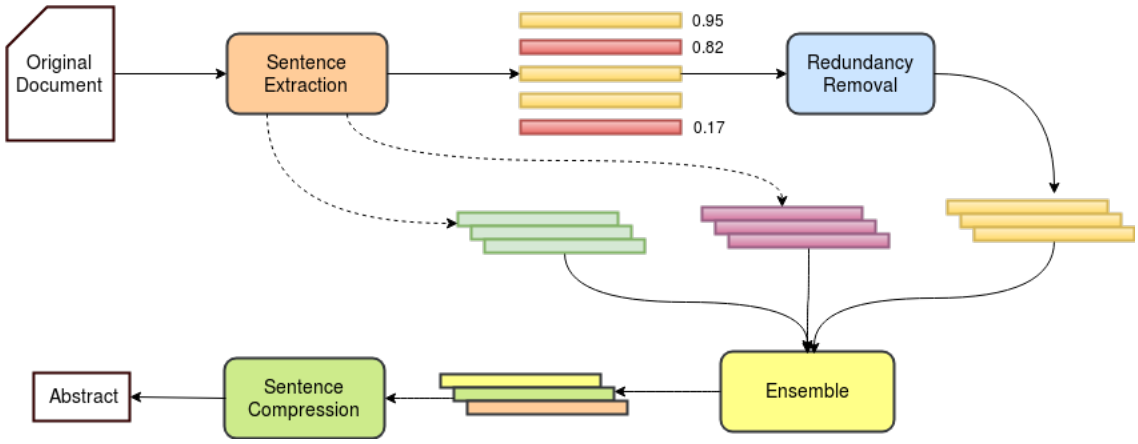

Figure 6.3: End to end abstractive summarization system

We use a total of 8 different sentence extraction techniques. Apart from the neural network based technique proposed in chapter 3, we use four commonly used methods mentioned in previous chapters, LexRank[20], TextRank[50], FreqSum[58] and TopicSum[13]. Beside this, we use YAKE, LCP and Legal boost techniques for the legal dataset. We also compare the results on the following ensemble systems:

- Graph1: WCS[80] based ensemble of extractive systems with the graph-based ranking algorithm and different similarity measures
- Centroid1: Same as Graph1 but with the centroid-based ranking algorithm
- Graph2: WCS based ensemble of extractive systems with the graph-based ranking algorithm and different text representation schemes. The text representation schemes for legal dataset include tf-idf, topic signatures, topics generated using YAKE and catchphrases using LCP[45]. The results for ACL corpus are based only on tf-idf and topic sum based representations
- Centroid2: Same as Graph2 but with the centroid-based ranking algorithm
- HybridRank: Ensemble of all the extractive techniques mentioned above, except the neural network based sentence extraction, using the HybridRank algorithm
- HybridRank + NN: Ensemble of all the extractive techniques mentioned above, including the neural network based sentence extraction, using the HybridRank algorithm

		Legal		ACL	
		Only SE	SE + SC	Only SE	SE + SC
Original + Sentence Comression	TextRank	0.16	0.17	0.018	0.022
	LexRank	0.155	0.162	0.02	0.023
	Freqsum	0.16	0.165	0.018	0.021
	TopicSum	0.178	0.185	0.02	0.026
	YAKE	0.159	0.167	-	-
	LCP	0.181	0.189	-	-
	LegalBoost	0.15	0.162	-	-
	Neural	0.215	0.27	0.027	0.032
Ensemble + Sentence Compression	Graph1	0.162	0.183	0.019	0.024
	Centroid1	0.165	0.18	0.018	0.019
	Graph2	0.179	0.194	0.02	0.026
	Centroid2	0.181	0.19	0.019	0.023
	HybridRank	0.187	0.199	0.022	0.028
	HybridRank+NN	0.219	0.281	0.023	0.028

Table 6.2: Results for end to end abstractive summarization system

As explained in chapter 3, ROUGE-4 makes more sense for the summaries generated on legal and acl corpora. Also, as evident from table 3.2 and 3.3, there is high co-relation between R1, R2 and R4 scores. Hence, for this experiment, we consider only the R-4 metric. Evaluation parameters were same as used in previous experiments. We report R-4 recall for Legal summaries and R-4 precision for the ACL summaries for a reason mentioned in chapter 3. Some of the original techniques like LCP, LegalBoost and YAKE are defined only for the Legal corpus.

The results of this end to end system are encouraging. By using the additional sentence compression module we were able to improve the *coverage drastically* of summaries for both Legal and ACL corpora. For Legal corpus, the ensemble HybridRank+NN gave best results. However, the difference between the 'HybridRnak+NN' and the 'Neural only' approach is not significant. In case of ACL corpus, the Neural approach outperforms the best performing ensemble by a small but significant margin. The results on ACL and Legal corpus are not comparable, since ACL is evaluated using R-4 precision and Legal using R-4 Recall. Due to the limitations of sentence compression mentioned in the previous sections, these results should be taken with a grain of salt. The fact that we were able to reproduce only 20 percent of sentence compressions for Legal data, and even lesser of the scientific articles, but achieve a significant improvement in ROUGE score highlights

the limitation of ROUGE metrics in estimating the summary quality. With data-driven techniques getting more popular, and the scale of evaluation increasing drastically, it is more important than ever to explore alternative evaluation measures which can work on a large scale. ROUGE scores, accompanied by a qualitative analysis on a small subset of the evaluation data is prone to bias and not a reliable alternate any longer. There can be few very good sentence compressions and generated sentences, and a very high ROUGE score, but it does not necessarily highlight the success of an overall system.

CHAPTER 7

Conclusion and future work

In this chapter we present an overview of the work that we discussed throughout the thesis and point out to some open questions and possible research directions. We proposed several techniques that can improve or compliment the existing sentence extraction systems. We introduced two new corpus consisting of Legal and scientific articles that can be used for evaluating sentence compression and abstractive summarization systems. We then proposed a attention model based sentence extraction technique that is capable of identifying key information from the documents, without requiring any manually labelled data. We showed that such techniques that use large number of pseudo-labelled data can easily outperform the systems that use domain knowledge and manual annotations.

Next we presented two approaches that can be used to combine several existing sentence extractors and generate a much better meta-system. The first system uses multiple sentence similarity metrics, ranking algorithms and text representation schemes to improve rank aggregation. In the other system we exploit similarities between summaries generated by several systems to estimate their reliability. We presented a hypothesis that the *Good* summaries are bound to share more content. We then use this reliability measure to weight the individual candidates when performing rank aggregation. Such a system appears to be very promising and is another highlight of how pseudo-labelled data can be used for improving the existing systems. We would like to highlight that a domain specific sentence extractor performs much better than an ensemble of generic systems. The sentence extractor trained on a legal dataset achieved a performance that was much higher than all the ensemble systems. We then presented a sentence compression technique that uses sequence to sequence model. We proposed a new encoder that can directly point out rare but important entities to the attention module and decoder.

We would like to conclude by pointing out several open questions that are worth exploring. As stressed several times throughout the thesis, the focus of research in summarisation, especially abstractive techniques, has been limited to newswire corpora. However, for most of these works, it is difficult to estimate how useful the system actually is. For instance, an example given in [68] shows the following:

(Origignal): A detained iranian-american academic accused of acting against national security has been released from a tehran prison after a hefty bail was posted, a to p judiciary official said tuesday.

(Human): iranian-american academic held in tehran released on bail

(System1): detained iranian-american academic released from jail after posting bail

(System2): detained iranian-american academic released from prison after hefty bail

From a human perspective, summaries from system 1 and system 2 would be equally useful for most people. There is little advantage in spending efforts to improve the results from system1 to system2. Such an improvement might be of marginal interest to most users, who dont care about that specific details in a news headline. We believe that instead of trying to build a general purpose abstractive technique, the focus should instead be on domain-specific cases. In such cases, the utility of system to end user can play the role of evaluation measure. Not only that, for some domains sentence compression techniques can solve several related problems as well. For example legal sentences on an average are much longer and much more complex. A compression system that can replace difficult phrases by simpler vocabulary would be extremely useful. Moreover, our experiments show that restricting to certain domains results in very good performance even for simple systems.

The first step towards achieving this goal this could be creating large reusable corpora. Recently several works have reported encouraging results while using larger pseudo-labelled dataset in favour of smaller manually annotated corpora [18, 39]. The results reported in this thesis are also on similar lines[47]. Eliminating the dependency on explicit human annotation would remove the biggest bottleneck of the data-driven approaches. Improving the evaluation metrics to involve explicit or implicit human feedback is another very important direction that needs to be explored.

References

[1] A. Abu-Jbara and D. Radev. Coherent citation-based summarization of scientific papers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 500–509. Association for Computational Linguistics, 2011.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[3] M. Banko, V. O. Mittal, and M. J. Witbrock. Headline generation based on statistical translation. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 318–325. Association for Computational Linguistics, 2000.

[4] R. Barzilay and K. R. McKeown. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328, 2005.

[5] R. Brandow, K. Mitze, and L. F. Rau. Automatic condensation of electronic publications by sentence selection. *Information Processing & Management*, 31(5):675–685, 1995.

[6] R. Campos, V. Mangaravite, A. Pasquali, A. M. Jorge, C. Nunes, and A. Jatowt. A text feature based automatic keyword extraction method for single documents. In *European Conference on Information Retrieval*, pages 684–691. Springer, 2018.

[7] Z. Cao, W. Li, S. Li, F. Wei, and Y. Li. Attsum: Joint learning of focusing and summarization with neural attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 547–556, 2016.

[8] J. Cheng and M. Lapata. Neural summarization by extracting sentences and words. *arXiv preprint arXiv:1603.07252*, 2016.

[9] S. Chopra, M. Auli, and A. M. Rush. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of Human Language Technologies: The 2016 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2016.

[10] J. Clarke and M. Lapata. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research*, 31:399–429, 2008.

[11] T. A. Cohn and M. Lapata. Sentence compression as tree transduction. *Journal of Artificial Intelligence Research*, 34:637–674, 2009.

[12] J. M. Conroy, J. D. Schlesinger, J. Goldstein, and D. P. OâĂŹleary. Left-brain/right-brain multi-document summarization. In *n Proceedings of the Document Understanding Conference (DUC) 2004.*

[13] J. M. Conroy, J. D. Schlesinger, and D. P. O'Leary. Topic-focused multi-document summarization using an approximate oracle score. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 152–159. Association for Computational Linguistics, 2006.

[14] H. T. Dang. Overview of duc 2005. In *Proceedings of the document understanding conference*, volume 2005, pages 1–12, 2005.

[15] D. Das and A. F. Martins. A survey on automatic text summarization. *Literature Survey for the Language and Statistics II course at CMU*, 4:192–195, 2007.

[16] H. Daumé III and D. Marcu. A noisy-channel model for document compression. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 449–456. Association for Computational Linguistics, 2002.

[17] S. T. Davis, J. M. Conroy, and J. D. Schlesinger. Occams–an optimal combinatorial covering algorithm for multi-document summarization. In *Data Mining Workshops*

(ICDMW), 2012 IEEE 12th International Conference on, pages 454–463. IEEE, 2012.

[18] M. Dehghani, H. Zamani, A. Severyn, J. Kamps, and W. B. Croft. Neural ranking models with weak supervision. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 65–74. ACM, 2017.

[19] S. Dumais, G. Furnas, T. Landauer, S. Deerwester, S. Deerwester, et al. Latent semantic indexing. In *Proceedings of the Text Retrieval Conference*, 1995.

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

[21] K. Filippova, E. Alfonseca, C. Colmenares, L. Kaiser, and O. Vinyals. Sentence compression by deletion with lstms. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[22] K. Filippova and Y. Altun. Overcoming the lack of parallel data in sentence compression. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1481–1491, 2013.

[23] F. Galgani, P. Compton, and A. Hoffmann. Hauss: Incrementally building a summarizer combining multiple techniques. *International Journal of Human-Computer Studies*, 72(7):584–605, 2014.

[24] D. Gillick and B. Favre. A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, pages 10–18. Association for Computational Linguistics, 2009.

[25] D. Gillick, B. Favre, D. Hakkani-Tür, B. Bohnet, Y. Liu, and S. Xie. The icsi/utd summarization system at tac 2009.

[26] C. Grover, B. Hachey, and C. Korycinski. Summarising legal texts: Sentential tense and argumentative roles. In *Proceedings of the HLT-NAACL 03 on Text summariza-*

tion workshop-Volume 5, pages 33–40. Association for Computational Linguistics, 2003.

[27] J. Gu, Z. Lu, H. Li, and V. O. Li. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1631–1640, 2016.

[28] C. Gulcehre, S. Ahn, R. Nallapati, B. Zhou, and Y. Bengio. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 140–149, 2016.

[29] A. Haghighi and L. Vanderwende. Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370. Association for Computational Linguistics, 2009.

[30] K. Hirohata, N. Okazaki, S. Ananiadou, and M. Ishizuka. Identifying sections in scientific abstracts using conditional random fields. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*, 2008.

[31] K. Hong, J. M. Conroy, B. Favre, A. Kulesza, H. Lin, and A. Nenkova. A repository of state of the art and competitive baseline summaries for generic news summarization. In *Proceedings of Language Resources and Evaluation Conference*, pages 1608–1616, 2014.

[32] K. Hong, M. Marcus, and A. Nenkova. System combination for multi-document summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[33] K. Hong and A. Nenkova. Improving the estimation of word importance for news multi-document summarization. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 712–721, 2014.

[34] M. Kågebäck, O. Mogren, N. Tahmasebi, and D. Dubhashi. Extractive summarization using continuous vector space models. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)@ EACL*, pages 31–39. Citeseer, 2014.

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

[36] H. Kobayashi, M. Noguchi, and T. Yatsuka. Summarization based on embedding distributions. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1984–1989, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[37] A. Kulesza, B. Taskar, et al. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2–3):123–286, 2012.

[38] J. Kupiec, J. Pedersen, and F. Chen. A trainable document summarizer. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 68–73. ACM, 1995.

[39] C. Liang, J. Berant, Q. Le, K. D. Forbus, and N. Lao. Neural symbolic machines: Learning semantic parsers on freebase with weak supervision. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 23–33, 2017.

[40] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8. Association for Computational Linguistics, 2004.

[41] C.-Y. Lin and E. Hovy. The automated acquisition of topic signatures for text summarization. In *Proceedings of the 18th conference on Computational linguistics-Volume 1*, pages 495–501. Association for Computational Linguistics, 2000.

[42] H. Lin and J. Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational*

Linguistics: Human Language Technologies-Volume 1, pages 510–520. Association for Computational Linguistics, 2011.

[43] H. Lin and J. Bilmes. Learning mixtures of submodular shells with application to document summarization. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 479–490. AUAI Press, 2012.

[44] H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165, 1958.

[45] A. Mandal, K. Ghosh, A. Pal, and S. Ghosh. Automatic catchphrase identification from legal court case documents. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2187–2190. ACM, 2017.

[46] P. Mehta. From extractive to abstractive summarization: A journey. In *Proceedings of the ACL 2016 Student Research Workshop, Germany*, pages 100–106. ACL, 2016.

[47] P. Mehta and P. Majumder. Content based weighted consensus summarization. In *European Conference on Information Retrieval*, pages 787–793. Springer, 2018.

[48] P. Mehta and P. Majumder. Effective aggregation of various summarization techniques. *Information Processing & Management*, 54(2):145–158, 2018.

[49] Q. Mei and C. Zhai. Generating impact-based summaries for scientific literature. *Proceedings of ACL-08: HLT*, pages 816–824, 2008.

[50] R. Mihalcea and P. Tarau. Textrank: Bringing order into texts. In *Proceedings of Emperical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[51] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[52] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[53] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[54] M.-F. Moens and C. Uyttendaele. Automatic text structuring and categorization as a first step in summarizing legal cases. *Information Processing & Management*, 33(6):727–737, 1997.

[55] O. Mogren, M. Kågebäck, and D. Dubhashi. Extractive summarization by aggregating multiple similarities. In *Proceedings of Recent Advances In Natural Language Processing*, pages 451–457, 2015.

[56] R. Nallapati, B. Zhou, C. dos Santos, C. Gulcehre, and B. Xiang. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, 2016.

[57] A. Nenkova and K. McKeown. A survey of text summarization techniques. In *Mining text data*, pages 43–76. Springer, 2012.

[58] A. Nenkova, L. Vanderwende, and K. McKeown. A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 573–580. ACM, 2006.

[59] K. Owczarzak, J. M. Conroy, H. T. Dang, and A. Nenkova. An assessment of the accuracy of automatic evaluation in summarization. In *Proceedings of Workshop on Evaluation Metrics and System Comparison for Automatic Summarization*, pages 1–9. Association for Computational Linguistics, 2012.

[60] K. Owczarzak and H. T. Dang. Overview of the tac 2011 summarization track: Guided task and aesop task. In *Proceedings of the Text Analysis Conference (TAC 2011), Gaithersburg, Maryland, USA*, 2011.

[61] T. Oya, Y. Mehdad, G. Carenini, and R. Ng. A template-based abstractive meeting summarization: Leveraging summary and source text relationships. In *Proceedings*

of the 8th International Natural Language Generation Conference (INLG), pages 45–53, 2014.

[62] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[63] S. Palchowdhury, P. Majumder, D. Pal, A. Bandyopadhyay, and M. Mitra. Overview of fire 2011. In *Multilingual Information Access in South Asian Languages*, pages 1–12. Springer, 2013.

[64] Y. Pei, W. Yin, Q. Fan, and L. Huang. A supervised aggregation framework for multi-document summarization. In *Proceedings of 24th International Conference on Computational Linguistics: Technical Papers*, pages 2225–2242, 2012.

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

[66] V. Qazvinian, D. R. Radev, S. M. Mohammad, B. Dorr, D. Zajic, M. Whidby, and T. Moon. Generating extractive summaries of scientific paradigms. *Journal of Artificial Intelligence Research*, 46:165–201, 2013.

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

[68] A. M. Rush, S. Chopra, and J. Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[69] G. Salton, A. Singhal, M. Mitra, and C. Buckley. Automatic text structuring and summarization. *Information Processing & Management*, 33(2):193–207, 1997.

[70] E. SanJuan, F. Ibekwe-SanJuan, J.-M. Torres-Moreno, and P. Velázquez-Morales. Combining vector space model and multi word term extraction for semantic query

expansion. *Natural Language Processing and Information Systems*, pages 252–263, 2007.

[71] M. Saravanan and B. Ravindran. Identification of rhetorical roles for segmentation and summarization of a legal judgment. *Artificial Intelligence and Law*, 18(1):45–76, 2010.

[72] M. Saravanan, B. Ravindran, and S. Raman. Improving legal document summarization using graphical models. *Frontiers in Artificial Intelligence and Applications*, 152:51, 2006.

[73] M. Saravanan, B. Ravindran, and S. Raman. Using legal ontology for query enhancement in generating a document summary. *FRONTIERS IN ARTIFICIAL INTELLIGENCE AND APPLICATIONS*, 165:171, 2007.

[74] M. Saravanan, B. Ravindran, and S. Raman. Automatic identification of rhetorical roles using conditional random fields for legal document summarization. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*, 2008.

[75] J. Steinberger. Using latent semantic analysis in text summarization and summary evaluation. In *Proceedings of ISIMâĂŹ04*, pages 93–100, 2004.

[76] L. Suanmali, M. S. Binwahlan, and N. Salim. Sentence features fusion for text summarization using fuzzy logic. In *Hybrid Intelligent Systems, 2009. HIS'09. Ninth International Conference on*, volume 1, pages 142–146. IEEE, 2009.

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

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

[79] E. M. Voorhees. The trec robust retrieval track. In *ACM SIGIR Forum*, volume 39, pages 11–20. ACM, 2005.

[80] D. Wang and T. Li. Weighted consensus multi-document summarization. *Information Processing & Management*, 48(3):513–523, 2012.

[81] K. Woodsend, Y. Feng, and M. Lapata. Generation with quasi-synchronous grammar. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 513–523. Association for Computational Linguistics, 2010.

[82] W. Yin, H. Schütze, B. Xiang, and B. Zhou. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association of Computational Linguistics*, 4(1):259–272, 2016.

[83] Q. Zhou, N. Yang, F. Wei, and M. Zhou. Selective encoding for abstractive sentence summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1095–1104, 2017.

CHAPTER A

The Dictionary built using LegalBoost Method

We list below a sample of the dictionary for legal domain that was built using LegalBoost technique mentioned in chapter 3.

penal	impugned
government	judicial
magistrate	curtailment
criminal	defamation
pronouncement	imprisonment
respondent	affidavit
appellant	counsel
clause	dismissed
petitioner	jurisdiction
petition	proceeding
factual	evidence
constitution	premise
charge	commissioner
opinion	cognizance
court	dispute

CHAPTER B

Summaries generated using rank aggregation

Document d30047

Greedy-Cosine Summary (ROUGE-1 0.3995): Endeavour and its astronauts closed in Sunday to capture the first piece of the international space station the Russian made Zarya control module that had to be connected to the Unity chamber aboard the shuttle. Their No. The project would allow Russia to keep up to 80 000 jobs over the next 15 years. The unmanned launch set for 9:40 Moscow time (0640 GMT) Friday from Baikonur in Kazakhstan has been delayed for one year mainly because of the cash strapped Russian space agency's failure to complete another part of the station. The 12.4 meter (41.2 feet) Zarya or Sunrise was launched by a three stage Proton booster rocket

---

Centroid Cosine Summary (ROUGE-1 0.4142):The international space station is a U.S. led successor to Russia's Mir space station and is being heralded by space officials as the vehicle for unprecedented international cooperation in space in the 21st century. Yuri Koptev head of the Russian Space Agency said the launching of Zarya a 42 000 pound space tugboat that will supply power and propulsion in the early stages of station construction was a needed vindication for the troubled Russian space program.WASHINGTON NASA and the Russian Space Agency have agreed to set aside a last minute Russian request to launch an international space station into an orbit closer to Mir officials announced Friday.

Graph-Cosine Summary (ROUGE-1 0.4240): Endeavour and its astronauts closed in Sunday to capture the first piece of the international space station the Russian made Zarya control module that had to be connected to the Unity chamber aboard the shuttle. Two astronauts ventured back out on another spacewalk Wednesday to attach antennas to the

international space station under construction nearly 250 miles above Earth. But because of all the partner nations trusting each other and working together he said the international space station is going to be a reality. Solar power panels on the module were successfully deployed shortly after it reached its first orbit and all systems were working normally officials said.

---

*Rank aggregation ensemble using WCS (ROUGE-1 0.4608):*Endeavour and its astronauts closed in Sunday to capture the first piece of the international space station the Russian made Zarya control module that had to be connected to the Unity chamber aboard the shuttle. WASHINGTON NASA and the Russian Space Agency have agreed to set aside a last minute Russian request to launch an international space station into an orbit closer to Mir officials announced Friday. The international space station is a U.S. led successor to Russia's Mir space station and is being heralded by space officials as the vehicle for unprecedented international cooperation in space in the 21st century.

Document d30029

Greedy-Cosine Summary (ROUGE-1 0.3357): Two sailors died and 15 others were missing after gale force winds and high seas battered yachts in the Sydney to Hobart yacht race Monday. Robin Poke a spokesman for the Australian Maritime Safety Authority (AMSA) admitted it was unlikely the 33 year old Charles could survive more than 24 hours at sea after being washed off Sword of Orion on Sunday night. Ellison says Sayonara can knock half a day off Morning Glory's record given the right conditions. (Barclay) came down into the water only about 5 10 feet away from me and I swam to him pretty quickly.

Greedy-Word Overlap Summary (ROUGE-1 0.3976): Two yacht crew members are dead three yachts remain missing and rescue resources were stretched to the limit Monday as huge seas and gale force winds continued to batter the Sydney to Hobart race fleet. Although the two were forced to reduce sail and cut speed they were still well inside the race record for the 630 nautical mile race of 2 days 14 hours seven minutes and 10 seconds set by German maxi Morning Glory in 1996. British sailor Glyn Charles was missing and

presumed drowned becoming the third fatality in the Sydney to Hobart yacht race while three others remained missing

---

Greedy-KL Summary (ROUGE-1 0.4143): Two sailors died and 15 others were missing after gale force winds and high seas battered yachts in the Sydney to Hobart yacht race Monday. British sailor Glyn Charles was swept off the Sword of Orion on Sunday night when the boat rolled in wild seas. With winds gusting to 90 mph (145 kph) and seas swelling to 35 feet (10 meters) the race continued even as rescue teams searched for the missing vessels. Although the two were forced to reduce sail and cut speed they were still well inside the race record of 2 days 14 hours 7 minutes 10 seconds

---

Rank aggregation ensemble using Borda (ROUGE-1 0.4313): Two sailors died and 15 others were missing after gale force winds and high seas battered yachts in the Sydney to Hobart yacht race Monday. With winds gusting to 80 knots and seas swelling to 10 meters (35 feet) the race continued even as rescue teams searched for the three missing yachts. A total of 37 yachts have been forced to retire from the 630 nautical mile race many having been dismasted or suffering injuries to crew. About half of the 115 yacht fleet have been forced out of the 1 160 kilometer (725 mile) race. There were unconfirmed reports that one of the dead sailors was British.

Document d31032

Greedy-Word overlap Summary (ROUGE-1 0.3603): The president and his doctors say Yeltsin has no serious health problems and will serve out the final two years of his term. Yeltsin 67 has a respiratory infection that forced him to cut short his first foreign visit in months on Monday. The court will take at least a week to consider the issue the Interfax news agency reported. Russian President Boris Yeltsin who is still recuperating from his latest illness has canceled a trip to an Asian summit next month his office said Friday. Yakushkin his spokesman reiterated Tuesday there was no talk about an early resignation. It's time for him to step aside.

Centroid-Word Overlap Summary (ROUGE-1 0.3701): Russian President Boris Yeltsin cut short a trip to Central Asia on Monday due to a respiratory infection that revived questions about his overall health and ability to lead Russia through a sustained economic crisis. Doctors ordered Russian President Boris Yeltsin to cut short his Central Asian trip because of a respiratory infection and he agreed to return home Monday a day earlier than planned officials said. Yeltsin has decided to send Prime Minister Yevgeny Primakov to the November summit of the Asia Pacific Economic Forum in Kuala Lumpur because it deals mostly with financial issues Yeltsin's office said.

Graph-Word Overlap Summary (ROUGE-1 0.3823): Russian President Boris Yeltsin cut short a trip to Central Asia on Monday due to a respiratory infection that revived questions about his overall health and ability to lead Russia through a sustained economic crisis. The 67 year old president whose health has often sidelined him during his seven years in power has spent most of the last two months out of the limelight and out of the Kremlin holding meetings in his country home outside Moscow. The president has been suffering from bronchitis and a cold this week which forced him to cut short a visit to Central Asia on Monday.

Rank aggregation ensemble using RR (ROUGE-1 0.4265): The president and his doctors say Yeltsin has no serious health problems and will serve out the final two years of his term. Russian President Boris Yeltsin cut short a trip to Central Asia on Monday due to a respiratory infection that revived questions about his overall health and ability to lead Russia through a sustained economic crisis. Yeltsin's resignation is highly unlikely. But Luzhkov has always been supportive of Yeltsin and had never before questioned the president's health or ability to govern. Doctors ordered Russian President Boris Yeltsin to cut short his Central Asian trip because of a respiratory infection and he agreed to return home

---

CHAPTER C

Visualizing sentence compressions on Legal data

- The underlined parts in the original sentence are the ones that are correctly omitted in the target sentence. The striked out part in the original sentences are wrongly missing in the translation, affecting the comprehensibility of the sentence.
- The striked out parts in the compressed sentence are the ones that should exist but are misplaced in the compressed sentence. Boldfaced parts in the compressed sentence are the ones newly introduced.
- The boldfaced parts in the expected compressions are the corrections that are made compared to the actual compression.

Original:

There can **in my opinion** be no escape from the conclusion that section 12 of the Act by which **a most important protection or** safeguard conferred on the subject by the Constitution has been taken away is not **a valid provision** since it **contravenes** the very provision in the Constitution under which the Parliament derived its competence to enact it.

Compressed:

There can be no escape from the conclusion that section 12 of the Act by which safeguard conferred on the subject by the Constitution has been taken away is not valid since it contravened the very provision in the Constitution under which the Parliament derived its competence to enact it.

Original:

The Act provides for levy of **two kinds of taxes called** the general tax and the special tax by the two charging sections 5 and 10 respectively. ~~**Seervai attempted to make out**~~ that the provisions of the charging sections 5 and 10 fixing Rs. 30000 and Rs. 5000 as the minimum taxable turnover for general tax and special tax respectively were found discriminatory and void under article 14 read with article 13 of the Constitution and he gave us several tables of figures showing how the imposition of the tax actually works **out in practice in hypothetical cases**.

Compressed:

The Act provides for the levy of the general tax and special tax by the two charging sections 5 and 10 respectively. that the provisions of the charging sections 5 and 10 fixing Rs. 30000 and Rs. 5000 as the minimum taxable turnover for general tax and special tax respectively are discriminatory and void under ~~**art**~~ of the Constitution and he gave the several tables of figures showing how the imposition of the tax actually works.

Expected Compression:

The Act provides for the levy of the general tax and special tax by the two charging sections 5 and 10 respectively. **Seervai attempted to make out** that the provisions of the charging sections 5 and 10 fixing Rs. 30000 and Rs. 5000 as the minimum taxable turnover for general tax and special tax respectively are discriminatory and void under **article 14 read with article 13** of the Constitution and he gave the several tables of figures showing how the imposition of the tax actually works.

Original:

The **learned trial** magistrate **believed the prosecution evidence rejected the pleas raised by the defence convicted** the appellants of the charge framed and sentenced them to undergo simple imprisonment for two months each. ~~**The appellate court**~~ confirmed the conviction of the appellants but reduced their sentence from simple imprisonment for two months to a fine of Rs. 50 or in default simple imprisonment for one month each.

Compressed:

The Magistrate **found** the appellants of the charge framed and sentenced them to undergo simple imprisonment for two months ~~**guilty**~~. confirmed the conviction of the appellants but reduced their sentence from simple imprisonment for two months to a fine of Rs. 50 or in default simple imprisonment for one month each.

Expected Compression:

The Magistrate found the appellants **guilty** of the charge framed and sentenced them to undergo simple imprisonment for two months. **The appellate court** confirmed the conviction of the appellants but reduced their sentence from simple imprisonment for two months to a fine of Rs. 50 or in default simple imprisonment for one month each.

CHAPTER D

List of Publications

Book

- P. Mehta and P. Majumder. From Extractive to Abstractive Summarization: A Journey. (Proposal accepted by Springer Nature)

Journal

- P. Mehta and P. Majumder. Effective aggregation of various summarization techniques. Information Processing & Management, 54(2):145-158, 2018.
- P. Mehta, P. Majumder. Large scale quantitative analysis of three Indo-Aryan languages. Journal of Quantitative Linguistics, 23(1):109-32, 2016

Conference and Workshops

- P. Mehta and P. Majumder. Content based weighted consensus summarization. In European Conference on Information Retrieval, pages 787-793. Springer, 2018.
- P. Mehta. From extractive to abstractive summarization: A journey. In Proceedings of the ACL 2016 Student Research Workshop, pages 100-106, Berlin, Germany, August 2016. Association for Computational Linguistics.
- Y. Keswani, H. Trivedi, P. Mehta, and P. Majumder. Author Masking through Translation-Notebook for PAN at CLEF 2016. In Conference and Labs of the Evaluation Forum, CLEF 2016

Under preparation:

- P. Mehta and P. Majumder. Exploiting local and global performance of candidate systems for aggregation of summarization techniques. arXiv preprint arXiv:1809.02343
- P. Mehta, G. Arora and P. Majumder. Attention based Sentence Extraction from Scientific Articles using Pseudo-Labeled data. arXiv preprint arXiv:1802.04675.