A Study on Neural Network Modeling Techniques for Automatic Document Summarization

Chun-I Tsai
Abstract

A Study on Neural Network Modeling Techniques for Automatic Document Summarization

Chun-I Tsai

With the Internet becoming widespread, countless articles and multimedia content have been filled in our daily life. How to effectively acquire the knowledge we seek becomes one of the unavoidable issues. To help people to browse the main theme of the document faster, many studies are dedicated to automatic document summarization, which aims to condense one or more documents into a short text yet still keep its essential content as much as possible. Automatic document summarization can be categorized into extractive and abstractive. Extractive summarization selects the most relevant set of sentences to a target ratio and assemble them into a concise summary. On the other hand, abstractive summarization produces an abstract after understanding the key concept of a document. The recent past has seen a surge of interest in developing deep neural network-based supervised methods for both types of automatic summarization. This thesis presents a continuation of this line and exploit two kinds of frameworks, which integrate convolutional neural network (CNN), long short-term memory (LSTM) and multilayer perceptron (MLP) for extractive speech summarization. The empirical results seem to demonstrate the effectiveness of neural summarizers when compared with other conventional supervised methods. Finally, to further explore the ability of neural networks, we experiment and analyze the results of applying sequence-to-sequence neural networks for abstractive summarization.

Key word: Automatic Document Summarization, Convolutional Neural Network, Long Short-Term Memory, Deep Neural Network, Sequence to Sequence Neural Network.
TABLE OF CONTENTS

LIST OF TABLES ........................................................................................................ iii
LIST OF FIGURES ....................................................................................................... iv

Introduction ................................................................................................................ 1
  1.1 Motivation .............................................................................................................. 1
  1.2 Research Issues and Contributions ....................................................................... 3
  1.3 Outline of the thesis .............................................................................................. 6

Related Work .............................................................................................................. 7
  2.1 Categorization of automatic summarization ....................................................... 7
  Figure 2.1. Categories of automatic summarization .................................................. 8
  2.2 Brief History of Automatic Summarization ....................................................... 9
    2.2.1 Text summarization ....................................................................................... 9
    2.2.2 Speech summarization ................................................................................ 13
  2.3 Classic methods for extractive summarization .................................................. 14
    2.3.1 Unsupervised methods ................................................................................ 15
    2.3.2 Supervised methods .................................................................................... 16
  2.4 Deep neural network based methods ................................................................ 19

Neural Network Based Methods for Summarization .............................................. 22
  3.1 CNN based summarizer ....................................................................................... 22
    Figure 3.1. The architecture of a convolutional neural network based modeling
    framework for extractive summarization ............................................................... 23
  3.2 CNN-LSTM based summarizer ......................................................................... 25
    Figure 3.2. The architecture of a CNN-LSTM based modeling framework for extractive
    LSTM based summarization ............................................................................... 25
    Figure 3.3. The architecture of neural summarizer for abstractive summarization... 28
  3.3 Abstractive neural summarizer ......................................................................... 28

Experimental Setup ............................................................................................... 32
  4.1 Speech and language corpora ............................................................................. 32
    4.1.1 Corpus for extractive summarization ......................................................... 32

Table 4.1. The statistical information of the broadcast news documents used in the
extractive summarization experiments. .................................................................... 33
    4.1.2 Corpus for abstractive summarization ....................................................... 33
  4.2 Evaluation metrics ............................................................................................. 34
Table 4.2. The agreement among the subjects for important sentence ranking for the evaluation set of MATBN corpus. .......................................................... 35
4.3 Features ........................................................................................................ 35
Figure 4.1 Similarities of top-N words in different embedding spaces .................. 36
4.3.1 Word embedding features ........................................................................ 36
Figure 4.2. Similarities of top-N words in different embedding spaces .................. 37
Figure 4.3. Relation of top-20 overlapping words and words frequencies............. 38
4.3.2 Traditional features .................................................................................... 38
Table 4.3. Typical features used to characterize spoken documents and their constituent sentences.................................................................................................................. 39
Experimental Results .......................................................................................... 42
5.1 Baseline experiment for extractive summarization ....................................... 42
5.1.1 classic methods with traditional features ............................................... 42
Table 5.1. Extractive Summarization results achieved by various state-of-the-art summarization methods with traditional features. ........................................ 43
5.1.2 Point-wise methods combined with word embedding feature .................... 43
Table 5.2. Summarization results achieved by point-wise supervised summarization methods combined with word embedding. ........................................... 44
5.2 Experiments on the proposed neural summarizers for extractive summarization ... 44
5.2.1 CNN based summarizer ............................................................................. 44
Table 5.3. Summarization results achieved by two CNN-based summarization methods .................................................................................................................... 45
5.2.2 CNN-LSTM based summarizer .................................................................. 45
Table 5.4. Summarization results achieved by leveraging different word embedding methods in the CNN-LSTM summarization framework.......................... 46
Table 5.5. Summarization results achieved by further incorporating the typical features in the CNN-LSTM summarization framework...................................... 46
Table 5.6. Summarization results achieved by abstractive summarizers .......... 48
5.2.3 Abstractive neural summarizer .................................................................. 49
Conclusion and Future Work............................................................................... 51
Bibliography .......................................................................................................... 53
# LIST OF TABLES

Table 4.1. The statistical information of the broadcast news documents used in the extractive summarization experiments. .................................................................33
Table 4.2. The agreement among the subjects for important sentence ranking for the evaluation set of MATBN corpus. .................................................................35
Table 4.3. Typical features used to characterize spoken documents and their constituent sentences. ........................................................................................................39
Table 5.1. Extractive Summarization results achieved by various state-of-the-art summarization methods with traditional features. .................................................43
Table 5.2. Summarization results achieved by point-wise supervised summarization methods combined with word embedding. .......................................................44
Table 5.3. Summarization results achieved by two CNN-based summarization methods. ..................................................................................................................45
Table 5.4. Summarization results achieved by leveraging different word embedding methods in the CNN-LSTM summarization framework. .................................46
Table 5.5. Summarization results achieved by further incorporating the typical features in the CNN-LSTM summarization framework ...........................................46
Table 5.6. Summarization results achieved by abstractive summarizers ........................48
LIST OF FIGURES

Figure 2.1. Categories of automatic summarization ......................................................... 8
Figure 2.2. A brief history of text summarization ................................................................. 12
Figure 3.1. The architecture of a convolutional neural network based modeling framework for extractive summarization ................................................................. 23
Figure 3.2. The architecture of a CNN-LSTM based modeling framework for extractive summarization ............................................................................... 26
Figure 3.3. The architecture of neural summarizer for abstractive summarization .......... 28
Figure 4.1 Similarities of top-N words in different embedding spaces ................................. 36
Figure 4.2. Similarities of top-N words in different embedding spaces ............................... 37
Figure 4.3. Relation of top-20 overlapping words and words frequencies ...................... 38
Figure 5.1 Comparison of abstractive summarization models with ground truth ........ 49
CHAPTER 1
Introduction

1.1 Motivation
With the rapid development of various Internet applications, unprecedented volumes of text documents and multimedia, such as broadcast radio, and television programs, lecture recordings, digital archives, among others, have been made available and become an integral part of our everyday life (Furui et al., 2012; Mari, 2008; Lee & Chen, 2005). The result is information overflow and makes useful knowledge hard to be found. It is impossible to digest every content of documents or videos even in a specific subject. To sufficiently leverage years of experience and wisdom of our human beings, there is an urgent need for an effective way to manage textual information and multimedia content.

Automatic document summarization is a proper direction to reduce the tedious process of understanding every document, which makes people understand the main concept of an article faster. Text summarization has captured researchers’ attention since sixty years ago. In late 1950s Luhn, (1958) (Simonyan & Zisserman, 2015) proposed the first algorithm for automatic summarization. Because high-frequency words are usually related to the topic of an article, Luhn’s, (1958) algorithm aims to select important sentences as summary according to different word frequency in each sentence. On the other hand, multimedia summarization, which input or output resource consists of images, video, music or speech, is a growing issue in recent year. Obviously, speech is one of the most important sources of information about multimedia. Explore technology about spoken document summarization can benefit multimedia summarization, since an utterance is formulated by our strength, emotion
and intention. By leveraging automatic speech recognition system and analyzing properties from a given speech signal like pitch, duration and energy, we can find clues to locate salient parts in an audio. In the past years, research tends to focus on extractive summarization in contrast to abstractive summarization. Extractive summarization methods aim to select the most important set of sentences from the original document as summary. Therefore, most of them reformulate the problem as finding a ranking function for each sentence in a document, and much easier to build a functional system. On the other hand, abstractive summarization methods generate a whole new summary after understanding the topic of the original document. These kinds of solutions require highly sophisticated knowledge about natural language processing and information retrieval, including semantic understanding and inference, information extraction and natural language generation. Therefore, constructing abstractive summarizer is one of the hardest challenges in automatic summarization.

Recently, deep learning- or deep neural network-based methods show powerful abilities in different domains (Hinton, et al., 2012; Simonyan & Zisserman, 2014). Beside reaching many state of the art classification performances, deep neural network also demonstrates amazing results on representation learning. This is a data driven technique that produces indicative features automatically during training, and significantly reduces efforts on heuristic feature engineering. This thesis presents a continuation of this line of research for automatic document summarization. We explore two different frameworks for extractive summarization, and experiment the performances on both speech and text documents. On the other hand, sequence to sequence neural network with attention mechanism shed the light on generating a smooth abstractive summarization. To understand what have been learned by such kind of network, we investigate the result of the latest abstractive summarizer.
1.2 Research Issues and Contributions

This thesis first investigates on extractive spoken document summarization, which manages to select a set of indicative sentences from a spoken document according to a target summarization ratio and concatenating them together to form a concise summary. Generally, methods for extractive summarization fall into two broad categories. One is based on structures of documents or sentence location information, and the other is sentence classification or ranking. Although most methods for extractive summarization are both suitable for text and spoken documents. There are several difficult issues specifically in the latter and make the performance of an applied text summarization method decreasing significantly on spoken documents. We list some of challenges for spoken document summarization below.

- Unlike text documents, which can be split into paragraphs, sentences and word boundaries exactly, spoken documents, however, usually lack such well-formed structures. Spoken documents are composed from a sequence of signals without these boundaries. Using “Pause” signal to split parts of summarization unit, for example, sentences is a common way. Several methods are used to handle this problem like Hidden Markov Model (HMM), Maximum Entropy and Conditional Random Field (CRF) (Liu, et al., 2006). However, determining boundaries is still a non-trivial task (Ostendorf, et al., 2008).

- Utilizing automatic speech recognition (ASR), a stream of speech signals can be transferred to text-form and then conventional text summarization technologies are introduced to distill salient information from the transcription of a spoken document. Nevertheless,
errors caused by misrecognition corrupt the quality of a transcription and result in summarization methods fail to work as well as expected. Because misrecognition errors (including insertion, substitution and deletion) apparently influence surface features like word frequency, number of named entities and structure of a sentence, or even worse, the meaning of original semantic flow on a higher level.

- Utterances naturally contain high-redundancy even without any misrecognition. Compared with written text documents, speech documents are full of grunts, function words, repeated or repaired words and restarted sentences. Despite breaking the structure of a sentence, these phenomena bring unrelated information into the whole document. Although some classic methods, for example, using a background language model or term frequency-inverse document frequency (TF-IDF) may alleviate this problem, but how to efficiently reduce disfluencies is an important topic for spoken document summarization.

There are still many problems not mentioned above, like overlapping speech and noise robustness. Besides, text normalization and label bias exist in both text and spoken document summarization. Interested readers may refer to (Liu & Hakkani-Tür, 2011) to a more detailed discussion.

Secondly, we explore the state-of-the-art methods on abstractive summarization, which do not copy indicative sentences completely from the original document but generates a rewritten summary (Ono, et al., 1994; Paice, 1990). Some primary methods like sentence compression or sentence fusion are cornerstones to achieve abstractive summarization, because both of them construct a new sentence but without
introducing words not shown in original document. Abstractive summarization seems as the final milestone in automatic summarization. In the past, a well-formed abstractive summarization is usually generated by a delicate combination of multiple modules. One of classic combinations is using information extraction technique as extractor to capture salient contents, and nature language generation as text realizer module to product a grammatical sentence. However most abstractive summarization methods are complex system built by several components and many heuristics are used in the past. It results in those systems lose generalities and are hard to maintain. Therefore, how to develop an effective method without so many exhausted experts’ rules and suitable for different scenarios is one of the most important issues for abstractive summarization.

In this thesis, we wish to use the least human effort to produce reliable results on both extractive and abstractive summarizations. Firstly, we experiment how deep learning approaches encode representations merely based on words yet still outperform empirical features on extractive text summarization. By taking advantages of convolutional neural network, which has shown remarkable ability on learning a sentence representation in a hyperspace, we further extract sentence-document pair features during training without exhausted feature engineering. To bridge the gap between text document and spoken document, we introduce Long Short-Term Memory (LSTM) to capture semantic flow on the sentence level so as to prevent our model from misleading of ASR errors. Secondly, we analyze influences of different word representation techniques. Finally, we investigate the latest method on abstractive summarization, which is constructed by LSTM encoder and decoder with the attention mechanism. This framework has been proven can automatically learn probability distribution of translating a word to another language even without any
predefined rule. We demonstrate the neural summarization is able to compress the meaning of multiple words and generate a word not featured in current article for abstractive summarization. The results show our neural summarization methods outperform classic methods with traditional features for extractive spoken document. On the other hand, different from previous assumptions, the abstractive summarization method we used effectively produces grammatical summary by an end-to-end module which is not a cascaded system.

1.3 Outline of the thesis

The remainder of this thesis is organized as follows:

- **Chapter 2**: We provide an overview of the research history of automatic summarization, including classic methods and state-of-the-art approaches which leverage deep neural networks for automatic summarization.

- **Chapter 3**: We present CNN based methods for extractive summarization on both text and speech corpora and explore an attention-based sequence-to-sequence network for abstractive text summarization.

- **Chapter 4**: We describe corpus, evaluation metrics, features and experimental setup.

- **Chapter 5**: We present experimental results of baseline and our methods.

- **Chapter 6**: We conclude this thesis and discuss possible directions of future works about these promising machine learning methods.

**CHAPTER 2**
In this chapter, we firstly go through how to categorize automatic summarization according to different sets of criteria. Second, we provide a history of research spectrum briefly. Third, we review several classic approaches for extractive summarization. Finally, the chapter ends with an introduction of neural network based methods in past few years.

2.1 Categorization of automatic summarization
In general, automatic summarization can be categorized by the following four aspects, namely, 1) input sources, 2) purposes, 3) functions and 4) methods. (Mani & Maybury, 1999)

- **Input Source:** In general, a summary is usually produced from a single document (single-document summarization), yet it can also be generated from a cluster of relevant documents, namely multiple documents (multi-document summarization). The latter suffer from two problems, that are information redundancy (Carbonell & Goldstein, 1998) and event ordering (or causality) (Kuo & Chen, 2008), because information is extracted from several documents.

- **Purposes:** Summary can be either generic or query-oriented. In generic summarization, each summary ignores user’s information needs and provides a general point of view of the document(s). On the contrary, query-oriented summarization is produced according to a specific topic or keyword form user’s concerns.
Functions: A summary can be informative, indicative or critical. An informative summary aims to provide a condensed presentation which details the main topics of the original document(s). It usually formulates a shorten version for original document(s). On the other hand, an indicative summary outlines the themes of original document(s), but does not discuss further information of original document(s). Finally, apart from two summary types, critical summary gives a judgement (positive and negative) on the input document(s). Although the critical summary might be not as objective as other types, it still attracts substantial research interest. (Galley et al., 2004)

Summarization Methods: There are several methods to produce a summary, such as extractive summarization, sentence compress, sentence fusion or abstractive summarization. First, extractive summarization composes a summary by selecting indicative sentences in the original document(s). Second, sentence compression removes
redundant words in each sentence. Third, sentence fusion gathers important words in the original document(s), and reconstructs a new set of sentences without introducing words out of the original document(s). Sentence compression and sentence fusion bridge the gap between extractive summarization and the last, abstractive summarization, which aims to generate a fluent and concise abstract by extracting important part and then phrasing the original document(s).

The categories of automatic summarization on different instantiations graphically displayed in Figure 2.1.

2.2 Brief History of Automatic Summarization

2.2.1 Text summarization

Text summarization has captured researchers’ eyes since 1950s. Early development in automatic summarization for the text document(s) is by pioneers such as Peter Luhn and Phyllis Baxendale. In 1958, Luhn Proposed a method, which models each sentence in the text document based on term frequency (TF) for extractive summarization (Luhn, 1958). In the paper, Luhn specifies a sentence as being constructed by function words and common words. The former make sentence fluent but does not affect meaning, and the latter reflect the central topic of sentence. Therefore, Luhn introduced inverse sentence frequency (ISF) to make common words and function words have proper weights of importance. Besides, since a sentence is represented by a high dimensional vector with dictionary number, stemming each word in sentence to root form so as to reduce dimensions is an essential part in the method. In the same year, (Baxendale, 1958) indicated the importance of position information. An observation from manually examining 200 individual paragraphs,
about 85 percent of the first sentence in each document reveals the main topic or contain the salient theme, and 7 percent concludes the document in the last paragraph. After a decade, Edmundson proposed two significant features, cue-word feature and title-word feature instead using term frequency and position information only (Edmundson, 1969). Cue-word, such as “in conclusion”, “impossible” and “significant” are those words do not contain the topic of the document, yet might indicate current or neighboring sentences are part of summary with high probability. Title-word is inspired by domain specific words linked with main topic usually present in the title. Therefore, A sentence contains title-words are more likely to be an indicative sentence. Edmundson furtherer proposed an algorithm combing Luhn’s and Baxendale’s, features namely, TF-ISF, position information, cue-word and title-word to select salient sentences, and the results shows effectiveness of those features. Thanks to research efforts of the three pioneers, which give a considerable influence in research community, the cornerstone of automatic summarization research has been well established.

After that, researchers kept introducing linguistic knowledge to automatic summarization in 1970. For example, Earl, (1970) assumed structure of a sentence is one of cues to define importance of sentence and leveraged part of speech (POS) to provide the structure information of each sentence. On the other hand, increasing interest in artificial intelligence encourages researchers to explore application of production rules, semantic network and for logic automatic text summarization in 1980s. In 1990s, following the success of statistical approaches in other natural language processing tasks, there was a trend to leverage statistical or machine learning methods in text automatic summarization. Such as in 1995, Kupiec proposed trainable method using Bayesian classification with uppercase words and sentence length
features (Kupiec et al., 1995). Because this kind of machine learning techniques can be adapted to different scenarios easily, a flourishing development has been studied in this aspect, like Bayesian classification (Kupiec et al., 1995), decision tree (Neto et al., 2002), constructive neural network (Chuang & Jihoon, 2000), Hidden Markov Model (HMM) (Mittendorf & Schäuble, 1994) and Support Vector Machine (Cao et al., 2006), to name just a few.

Although extractive summarization has been studied for many years, abstractive summarization has not seen the same amount of focus from the research community. Several methods are proposed yet still limited in a small domain and using exhausting generation rules which are far away from well developed. For example, in 1998, Eduard proposed a nature language generation (NLG) method composed by multiple modules which deals topic identification, interpretation and sentence generation individually (Eduard & Lin, 1998). However, the generation module is using an off-the-shelf method and is not really appropriate to summarization. In 2002, Saggion further proposed a summarization system which is able to produce indicative or informative abstract for multi-document. This method is also developed based on shallow syntactic and semantic analysis which are manually acquired by observation and testing (Saggion & Lapalme, 2002). White proposed an information extraction based abstractive summarizer, which requires a set of documents to be extracted, preferred scenario template and filters in 2001 (White, et al., 2001). Because it seemed too hard to achieve a significant breakthrough on abstractive summarization, some researchers turn attention to intermediate methods between extractive summarization. Sentence fusion is a promising strategy which has been a research task since 2005 (Barzilay & McKeown, 2005). This kind of method aims to eliminate redundancy by reconstructing a new sentence by only selecting words in original
In 2010, Filippova proposed an effective and unsupervised method that only using term frequency to formulate a word graph, and then producing a fused sentence according to POS-tagging and term co-occurrence (Filippova, 2010). Similar to Filippova’s method, Boudin developed a method leveraging key-phrase extraction, N-shortest paths algorithm and punctuated mark information in 2013 (Boudin & Morin, 2013). Although the results show Filippova’s method perform better at grammaticality, this method provides much more information than the other. Beside sentence fusion, sentence compression is another strategy, which deletes non-essential information word in each sentence yet still maintains grammatical correctness and the same number of sentences in original document (Yousfi-Monod, 2007). In 2000, Knight developed two approaches, decision tree and noisy-channel model to choose which words should drop or remain (Knight & Daniel, 2000). Sentence compression technique is also advantageous to speech summarization. In 2004, Hori demonstrated a dynamic programming method with term co-occurrence, tri-gram language model and confidence score features to select a set of salient words from a spoken document (Hori & Furui, 2004). Recently, deep learning, or deep neural network based methods has improved both extractive or abstractive summarization significantly, and a
detailed review about these methods will be given at the end of the chapter. The history of automatic summarization research spectrum is graphically depicted in Figure 2.2, which puts more emphasis on well-known statistical methods since the methods we proposed and explored are all belong to data-driven and supervised methods.

2.2.2 Speech summarization

Following the maturity of text summarization and growing number of available audio data, the research effort has been extended to speech summarization in parallel to the development of speech recognition (McKeown, Hirschberg, Galley, & Maskey, 2005). By virtue of speech summarization, one can efficiently digest multimedia content by listening to the associated speech segments of a summary. Several scenarios have been investigated, for example broadcast news (Maskey & Hirschberg, 2005; Crammer & Singer, 2002; Chen, Chen, & Wang, 2009), lectures (Zhang, et al., 2007), meetings (Murray, et al., 2005; Galley, 2006; Yang & Xie, 2008) or voicemail (Konstantinos & Renals, 2000). Since speech is formulated by emotionally, semantically and acoustically rich signals, a various set of features can be used to characterize a spoken document or sentence. Generally, spoken document features can be categorized from four aspects, namely, acoustic, structural, lexicon and relevance features (Galley, 2006; Zhang et al., 2007; Lin et al., 2009; Gerald & Zhu, 2008). First, acoustic features contain pitch level, energy, speech rate and speech intensity which might be changed when the speaker want to emphasize key point. Those prosodic cues are also the most obvious differences between speech summarization and text summarization. Second, structural features are usually a main concern of some specific types of spoken data, like broadcast news or lectures, which exhibit structure and tend to give an overview at the first part or conclude a summary in the end. Third,
lexicon features, which generated from ASR system can provide cues like language model scores, sentence length, number of named entity, stop words. However, because the transcriptions of a speech data are corrupted by ARS errors, the effectiveness of lexicon features decreases significantly when compared to text summarization. Finally, relevance features are a set of relations between sentences and the whole document. For instance, similarity between sentence and document measured from vector space model, or other unsupervised machine learning methods.

2.3 Classic methods for extractive summarization

The wide variety of extractive speech summarization methods that have been developed so far could be categorized into two broad groups, namely, unsupervised and supervised methods. In this section, we review both unsupervised supervised machine learning methods for extractive summarization that we use as our baseline in chapter 5. Without loss of generality, the sentence ranking strategy for extractive speech summarization can be stated as follows. Each sentence $S_i$ in a spoken document to be summarized is associated with a set of $M$ indicative features $x_i = \{x_{i1}, \ldots, x_{im}, \ldots, x_{iM}\}$ (usually, represented in vector form) and a summarizer (or a real-valued ranking function) is employed to assign a decision (or importance) score to each sentence $S_i$ according to its associated features $x_i$. Sentences of the document in turn can be ranked and iteratively selected to be included into the summary based on their scores until the length limitation or a desired summarization
ratio is reached. During the training phase, a set of training spoken documents $D = \{D_1, ..., D_n, ..., D_N\}$, consisting of $N$ documents and their corresponding handcrafted summary information, is used to train the supervised summarizer (or model).

2.3.1 Unsupervised methods

A common practice of most unsupervised methods is to select important sentences based on statistical features of sentences or of the words in the sentences, where the extraction of features and the training of corresponding models for sentence selection are typically conducted in the absence of human supervision. Statistical features, for example, can be the term (word) frequency, linguistic score and recognition confidence measure, as well as the prosodic information. Numerous unsupervised methods based on these features have been proposed and has sparked much interest recently. Among them, we choose two superior approaches, the integer linear programming (ILP) method (McDonald, 2007) and submodularity-based method (Lin & Bilmes, 2010).

ILP is designed to optimize the performance in a constrained situation, where both objective function and constraint are a set of integer variable and linear combined. When implemented to extractive summarization, ILP method perform a global optimization by maximizing the coverage of main concept of original document in a constrained summary length. Take the objective function in (McDonald, 2007) for example, which formulates extractive summarization problem as following:

$$\text{maximize } \sum_i \alpha_i \text{Relevance}(i) - \sum_{i<j} \alpha_{ij} \text{Redundancy}(i,j)$$

such that $\forall i, j$: (1)$\alpha_i, \alpha_{ij} \in \{0,1\}$

(2) $\sum_i \alpha_i l(i) \leq K$
\[(3)\alpha_{ij} - \alpha_i \leq 0\]
\[(4)\alpha_{ij} - \alpha_j \leq 0\]
\[(5)\alpha_i + \alpha_j + \alpha_{ij} \leq 1\]  
\[(2.1)\]

where \textit{Relevance}(i) is the similarity of sentence \(S_i\) to the whole document and \textit{Redundancy}(i,j) measures relevance between sentence pairs \(S_i\) and \(S_j\); \(\alpha_i\) and \(\alpha_{ij}\) are two indicator variable that are 1 when the sentence \(S_i\) or the sentence pairs \(S_i\) and \(S_j\) are part of summary; \(l(i)\) is the word number of sentence \(S_i\) and \(K\) is the constrained length of summary.

Submodular is a graphic method which inspired by budget constraint from economics. Submodularity-based method formulates summary length, one of neutrality of extractive summarization as budget constraint. Take the submodular function \(f(\cdot)\) in (Lin & Bilmes, 2010) for example, which can be stated formally as following:

\[
\max_{S \subseteq V} \{f(S); \sum_{i \in S} c_i \leq B\}
\]  
\[(2.2)\]

where \(V\) is the entire set of linguistic units, for example, sentences in the original document; \(S\) is the selected sentence; \(c_i\) is the non-negative cost of sentence \(S_i\) and \(B\) corresponds budget.

2.3.2 Supervised methods

Supervised methods need a set of training documents along with their corresponding human-annotated summaries for training their associated models. However, they in general lead to better summarization performance than the unsupervised methods. Here we introduce some of them, such as the support vector machine (SVM), Ranking support vector machine (Ranking SVM) and Global Conditional Log-linear Model (GCLM).
First, an SVM summarizer is developed under the basic principle of structural risk minimization (SRM) in the statistical learning theory (Vapnik & Vapnik, 1998). If the dataset is linear separable, SVM attempts to find an optimal hyper-plane by utilizing a decision function that can correctly separate the positive and negative samples, and ensure the margin is maximal. In the nonlinear separable case, SVM uses kernel functions or defines slack variables to transform the problem into a linear discrimination problem. We use the LIBSVM toolkit to construct a binary SVM summarizer, and adopt the radial basis function (RBF) as the kernel function. The posterior probability of a sentence $S_i$ being included in the summary class $S$ can be approximated by the following sigmoid operation:

$$P(S_i \in S|x_i) \approx \frac{1}{1+\exp(\alpha g(S_i)+\beta)}$$

(2.3)

where the weights $\alpha$ and $\beta$ are optimized by the training set, and $g(S_i)$ is the decision score of the sentence $S_i$ provided by the SVM summarizer. Once the SVM summarizer has been properly constructed, the sentences of a spoken document to be summarized can be ranked by their posterior probabilities of being in the summary class. The sentences with the highest probabilities are then selected and sequenced to form the final summary according to a predefined summarization ratio.

Parameters of SVM summarizer is optimized according to reducing the classification errors of the training sentences. It is anticipated that minimizing the classification errors caused by the summarizer would be equivalent to maximizing the lower bound of the summarization evaluation score (usually, the higher the score, the better the performance). In addition, the SVM summarizer in fact treats each training (summary or non-summary) sentence independently in estimating the corresponding model parameters. Theoretically, such a training paradigm can be referred to as point-wise learning.
Second, in contrast to SVM, Ranking SVM seeks to create a more rank- or preference-sensitive ranking function. It assumes there exists a set of ranks (or preferences) \( L = \{l_1, l_2, ..., l_K\} \) in the output space, while in the context of speech summarization, the value of \( K \), for example, can be simply set to 2 representing that a sentence can have the label of being either a summary \((l_1)\) or a non-summary \((l_2)\) sentence. The elements in the rank set have a total ordering relationship \( l_1 > l_2 > \cdots > l_K \), where \( > \) denotes a preference relationship. The training objective of Ranking SVM is to find a ranking function that can correctly determine the preference relation between any pair of sentences:

\[
S_i > S_j \iff f(S_i) > f(S_j)
\]

(2.4)

where \( l(S_i) \) denotes the label of a sentence \( S_i \) and \( f(S_i) \) denotes the decision value of \( S_i \) provided by Ranking SVM. As such, the corresponding training paradigm of Ranking SVM, on the other hand, is known as pair-wise learning.

Finally, The GCLM method has its roots from speech recognition for re-ranking recognition hypotheses for better recognition accuracy (Roark, et al., 2004; Roark, et al., 2007). The decision score that GCLM gives to a candidate summary sentence \( S_i \) can be computed by

\[
f(S_i) = \alpha \cdot x_i
\]

(2.5)

where \( x_i \) is the feature vector used to characterize a candidate summary sentence \( S_i \), and \( \alpha \) is the model parameter vector of GCLM. The model parameter vector \( \alpha \) is estimated by maximizing the following objective function:
\[ F_{GCLM}(\alpha) = \sum_{n=1}^{N} \sum_{S_R \in \text{Summ}_n} \log \frac{\exp(\alpha x_R)}{\sum_{S_J \in \text{D}_n} \exp(\alpha x_J)} \]  

(2.6)

By doing so, the GCLM method will maximize the posterior of the summary sentences (and thereby minimize the posterior of the non-summary sentences) of each given training spoken document.

2.4 Deep neural network based methods

The recent past has witnessed a flurry of research interest in developing deep learning- or deep neural network-based methods in many research societies and achieved extraordinary breakthroughs, such as speech recognition (Hinton, et al., 2012), image captioning (Xu, et al., 2015) and machine translation (Luong, Pham, & Manning, 2015). Following the success of deep learning based approaches, several methods were also developed for automatic summarization (Gupta & Lehal, 2010; Kågebäck, et al., 2014; Rush, et al., 2015). Similar to history of automatic summarization, researchers explored deep learning techniques on extractive summarization at first. For example, an intuitive way to implement deep neural network to extractive summarization is leveraging a multilayer perceptron (MLP) with multiple hidden layers, for important sentence selection. The decision score that the DNN-based method (e.g., with \( I \)-1 hidden layers) gives to a candidate summary sentence \( S_l \) can be computed by:

\[
\begin{align*}
\mathbf{h}^1 &= \alpha_1 (W_1 \mathbf{x}_i + \mathbf{b}_1) \\
\mathbf{h}^k &= \alpha_i (W_i \mathbf{h}^{k-1} + \mathbf{b}_i), \; k \in \{2, 3 \ldots K\} \\
P(S_l \in \mathbf{S} | \mathbf{x}_i) &\approx f(\mathbf{h}^K)
\end{align*}
\]

(2.7)

where \( \mathbf{x}_i \) is the feature vector used to characterize a candidate summary sentence \( S_l \); \( \mathbf{h}^k \) is \( k \)-th hidden vector; \( W_i \) and \( \mathbf{b}_i \) are the weight matrices and bias vectors of the
DNN model; \( f(\cdot) \) and \( \alpha_i(\cdot) \) are the element-wise activation functions for the output layer and the hidden layer \( i \), respectively (for example, the activation function can be a sigmoid function for the output layer and a hyperbolic tangent function for each hidden layer). Namely, in (2.7), a candidate summary sentence having a higher regression score (ranging between 0 and 1) is more likely to be selected into the summary.

Beside the basic framework like MLP, some classic neural models are implemented to automatic summarization. In 2015, (Chen, et al., 2015) proposed an unsupervised method for extractive broadcast news summarization which leverages recurrent neural network (RNN) language model. RNN is a neural model which can keep information in different time step and is good at handling sequence dependent problems. To be more specific, we can formulate the hidden layer of a RNN as below

\[
\mathbf{h}_t = \alpha(W_{hh}\mathbf{h}_{t-1} + W_{xh}\mathbf{x}_t + \mathbf{b})
\]

(2.8)

where \( \mathbf{h}_t \) corresponds the output of the hidden layer at time step \( t \); \( \alpha(\cdot) \) is the element-wise activation function just like MLP; \( W_{hh} \) is the recurrent weight to calculate the information kept in previous time steps; \( \mathbf{x}_t, W_{xh} \) and \( \mathbf{b} \) is the input vector at time step \( t \), the corresponded weight matrix to hidden layer and bias. The method proposed by (Chen, et al., 2015) produce a summary by ranking a set of sentence-document likelihoods generated from RNN language model. In 2015, (Cao, et al., 2015) developed a neural framework based on recursive neural network with a set of handcrafted features for multi-document summarization. Recursive neural network is different from RNN, which aims to deal input data through a parsing tree and rank each node as a hierarchical regression process. More recently, (Cheng & Lapata, 2016) proposed an exquisite framework which is constructed by CNN and RNN with attention mechanism. This method formulates summarization as a
sequential labeling problem. The first part of Cheng’s framework is an encoder, which starts with CNN to extract sentence feature, and then a cascaded RNN not only retain each high-level sentence vectors in previous time steps but compress the corresponding document vector in the end. After that, a sentence extractor takes account of sentence vector driven from CNN at previous module, document vector and the reminded sentence vector which kept in the RNN at the same time step to define whether labeling the sentence as one of summary. A similar idea can be found in Remesh’s work, which is constructed by two RNN modules, one encodes each sentence and the other labels the sentence belonging to summary or not according to the high-level representation of sentence (Nallapati, Zhai, et al., 2017). Beside generic summarization, Query-oriented summarization methods are also developed via deep learning in 2012 (Yan, Zhong, & Li, 2012) and 2017 (Yousefi-Azar & Hamey, 2017). In 2016, (Ren, Wei, Chen, Ma, & Zhou, 2016) further proposed a feedforward neural network and training algorithm, which tries to optimize ROUGE scores by eliminate redundancy between the selecting sentence and extracted summary. On the other hand, RNN with attention mechanism shed new light on abstractive summarization, that this kind of frameworks are able to generate a meaningful and grammatical correct sentence with minimal human effort. Inspired by RNN based architectures for machine translation, which generally contain three modules, encoder, decoder and attention module (Bahdanau, Cho, & Bengio, 2015), some methods are also proposed for abstractive summarization (Rush, Sumit, & Jason, 2015; Nallapati, et al., 2016). Although this kind of frameworks face problems, for instance, generating an ideal length of summary, unseen proper noun extraction or sentence repetition, they achieve impressive and promising results. In the chapter 3 of this thesis, we will explore a way to handle the problem of unseen proper noun extraction and sentence repetition.
CHAPTER 3
Neural Network Based Methods for Summarization

In this chapter, we present two kinds of neural network based framework for extractive summarization. We go through the process and detail functions of components of both framework individually.

3.1 CNN based summarizer

We first explore the use of a convolutional neural network (CNN) based modeling framework for extractive summarization, whose architecture is schematically depicted in Figure 3.1. At the first place, words within a document $D$ to be summarized and each of its constituent sentences $S$ are represented by distributed vectors, which can be derived with a large background document collection and a typical word embedding method such as the Skip-gram method (Mikolov, et al., 2013; Pennington, et al., 2014). Vectors of the words in $D$ represented in $e$ dimensions are adjoined in sequence to form a document word matrix $W_D \in \mathbb{R}^{e \times |D|}$ that is subsequently taken as the input to one of the two CNNs. The sentence word matrix $W_S \in \mathbb{R}^{e \times |S|}$, of a constituent sentence is likewise obtained and fed into the other CNN:

$$W_S = \begin{bmatrix} | & | & | \hline w_1 & \ldots & w_{|S|} \hline | & | & | \end{bmatrix}, \quad W_D = \begin{bmatrix} | & | & | \hline w_1 & \ldots & w_{|D|} \hline | & | & | \end{bmatrix}$$

(3.1)
Both of the two CNNs contain a convolutional layer and a max pooling layer; they in turn generate the low-dimensional embeddings (intermediate vector representations) of the document itself and each of its constituent sentences, respectively. To be more specific, a convolutional layer contains \( n \) filters with \( e \times m \) size \( f \in \mathbb{R}^{n \times e \times m} \), and each filter \( f \) multiplies the element-wise value with a (sentence or document) word matrix entirely to generates a corresponding feature map \( F \) in a vector form. Each component \( F_i \) in a feature map is as follows:

\[
F_i = \alpha(W \ast f)_i = \alpha(W_{i-m+1:i}^l \cdot f) = \alpha(\sum_{k=i}^{i+m-1} W_k f_k) \tag{3.2}
\]

Where \( W \) is a word matrix of sentence or document; \( \ast \) is the operator which computes an element-wise product between a column slice of \( W \) and the filter matrix \( f \); \( \alpha(\cdot) \) is activation function. Because the length of each sentence is different, the word vectors are padded with zeros when needed. Subsequently, a max pooling layer generates intermediate vector representation \( \mathbf{s} \) or \( \mathbf{d} \) for a sentence or document by extracting and cascading the maximum element from each feature maps:

\[
\mathbf{s} = [\max(F^1), \max(F^2), ..., \max(F^n)]
\]
\[ d = [\max(F^1), \max(F^2), \ldots, \max(F^n)] \]  

(3.3)

Where \( F^i \) and \( F^i' \), \( i \in \{1, 2, \ldots, n\} \) are sentence and document feature maps driven from CNN extractors. CNN have the ability to effectively capture the compositional process of mapping the meaning of individual words in a document or a sentence to a continuous representation of the document or the sentence, respectively.

A similarity matrix \( M \) is introduced here to facilitate the calculation of the similarity measure between the embeddings of the document and each of the document’s constituent sentences:

\[
sim(s, d) = s^\top M d
\]  

(3.4)

The two embeddings of a document-sentence pair and their similarity measure are in turn taken as the input to a multilayer perceptron (MLP) to induce a ranking score for each sentence. Apart from those derived from CNN, the input of MLP can be additionally augmented by a rich set of prosodic and lexical features \( x_{add} \), that characterize the relatedness between the document and each of its sentences or quantify the importance of each sentence. The probability of a sentence belonging to summary is computed as follows:

\[
x_{all} = [s; d; \sim(s, d); x_{add}]
\]

\[
P(S \in S|x_{all}) \approx f(W_1(\ldots \alpha_1 (W_1 x_{all} + b_1) \ldots ) + b_1)
\]  

(3.5)

The sentences with the highest probabilities (or regression scores) output by the CNN-based summarizer are then selected and sequenced to form the final summary according to different summarization ratios.

During the training phase, given a set of training spoken documents alongside their corresponding human-annotated (reference) summary sentences, the CNN-based summarizer is trained in the sense of reducing the regression errors of the summarizer.
made on the sentences of these training spoken document exemplars (ideally, the output of MLP should be “1” for summary sentences and “0” otherwise). Again, it is expected here that minimizing these errors caused by the CNN-based summarizer would be equivalent to maximizing the lower bound of the summarization evaluation value (usually, the higher the value, the better the performance).

More recently, methods with similar CNN-based modeling architectures have been applied with success on some question answering tasks (Bordes, et al., 2014; Severyn & Moschitti, 2015). However, as far as we know, this work is the first attempt to leverage such a modeling architecture for extractive speech summarization.

3.2 CNN-LSTM based summarizer

We further extend the CNN based framework in the previous section for automatic summarization. Obviously, the length of documents to be summarized are times of length of sentences. In order to bridge the gap, we assemble both CNN and LSTM networks and comes up a new component, document reader in this neural summarization framework.

We all agree upon that the CNN can extract local information effectively and the LSTM is good at capturing the long-term dependencies. These considerations motivate us to hybridize the advantages of CNN and LSTM networks so as to come up a new neural summarizer. Broadly, the architecture of the proposed framework consists of three modules: a sentence reader, a document reader and a summary sentence selector. The proposed architecture schematically is depicted in Figure 3.2.

More formally, at the first place, words within a document $D$ to be summarized and each of its constituent sentences $S$ are represented by distributed vectors, which can be derived with a large background document collection and a typical word
embedding method such as the Skip-gram method (Mikolov, et al., 2013) and the GloVe method (Pennington, et al., 2014). Vectors of the words in a constituent sentence of $D$ are adjoined in sequence to form a sentence matrix that is subsequently taken as the input to the sentence reader in the same way of CNNs based framework in the previous section. To preserve the local regularity information, the sentence reader is built by CNN. With the intension of summarizing the temporal variations of semantic themes of the whole document, all of the sentence matrices are fed into another CNN individually, and then an LSTM is stacked to sweep over all the sentences. In other words, the document reader, which is designed to encapsulate the semantic flow of the whole document, is composed of a CNN and an LSTM rather
than a single CNN. LSTM is similar to RNN, which can avoid the vanishing or exploding gradients problem during training-phase and keep long-term information more effectively. LSTM is implemented as the following:

\[
i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i)\\
\]

\[
f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f)\\
\]

\[
c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_i)\\
\]

\[
o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o)\\
\]

\[
h_t = o_t \tanh(c_t) \tag{3.6}\\
\]

Where \(x_t\) is the \(t^{th}\) sentence vector driven from CNN in document reader; \(\sigma\) and \(\tanh\) are logistic sigmoid function and hyperbolic tangent; \(i\), \(f\), \(o\) and \(c\) are input gate, forget gate, output gate and cell vectors; \(W_{xi}\) and \(b_i\) are the weight matrix and bias of input gate. Document reader encodes the intermediate vector representation \(d\) for a document at the last time step.

Two things are worthy to emphasize again. On one hand, CNN have the ability to effectively capture the compositional process of mapping the meaning of individual words in a sentence to a continuous representation of the sentence. In the proposed framework, both of the two CNN implement a convolutional layer and a max pooling layer; they in turn generate the low-dimensional embeddings (intermediate vector representations) at the sentence-level. On the other hand, for generating an enriched document representation, the LSTM encodes a sequence of sentences vectors, which are generated by CNN, into a document vector. The ability of LSTM, which is good to organize long-term relationship and co-occurrence, has been proven in may context-dependent tasks (Graves, Jaitly, & Mohamed, 2013).

Consequently, the two embeddings of a document-sentence pair are in turn taken as the input to a MLP, that is the summary sentence selector, to induce a ranking score
for each sentence. Apart from those derived from CNN-LSTM, we again take advantage of typical features and similarity matrix $M$ introduced above to score a sentence. The sentences with the highest probabilities (or regression scores) output by the neural summarizer is then selected and sequenced to form the final summary according to different summarization ratios.

3.3 Abstractive neural summarizer

In this section, we explore the promising neural sequence-to-sequence frameworks which are not a sentence selector but an abstract generator. Here we make use of the architecture proposed in 2016, whose framework is graphically displayed in the Figure 3.3 (See, Liu, & Manning, 2017). First, the bidirectional LSTM encoder generates and keeps encoder hidden vector $h_i$ of words within a document $D$ to be summarized. Note that gate units of bidirectional LSTM are similar to conventional one we mentioned above, yet bidirectional LSTM encodes $h_i$ both forward and backward. We can simply formulate $h_i$ as following:

![Figure 3.3. The architecture of neural summarizer for abstractive summarization](image-url)
\[
\tilde{h}_i = \tanh(W_{xh}x_i + W_{hh}h_{i-1} + b_h)
\]
\[
\tilde{h}_i = \tanh(W_{xh}x_i + W_{hh}h_{i+1} + b_h)
\]
\[
h_i = [W_{hh} \tilde{h}_i; W_{hh} \hat{h}_i] \quad (3.7)
\]

Where \(\tilde{h}_i\), \(x_i\), \(b_h\), \(W_{xh}\), \(W_{hh}\) and are \(i^{th}\) forward hidden vector, word embedding of \(w_i\), bias, weight matrix of word embedding vector and weight matrix of recurrent vector. \(h_i\) is concatenation of the reweighted \(\tilde{h}_i\) and \(\hat{h}_i\). The final output of the encoder, \(h_{|D|}\), is initialized as the recurrent vector \(d_{t=0}\) of the decoder, which is constructed by a single-directional LSTM. During training phase, the decoder is fed word embedding of previous word in the correct summary and generates a decoder hidden vector \(d_t\) at each time step \(t\), while in the testing phase it receives the word embedding from previous time step by decoder itself. Each \(d_{t|[t=1-|D|]}\) is used to compute correlation between all hidden vectors, \(h_i\) from encoder so as to obtain the attention distribution, which constrains model focus on a specific part of encoder hidden vectors:

\[
a^t_i = \mathbf{v}^\top \tanh(W_e h_i + W_d d_t + b_{att})
\]
\[
\tilde{a}^t = \text{softmax}(a^t)
\]
\[
\hat{a}^t_i = \frac{e^{a^t_i}}{\sum_{k=1}^{|D|} e^{a^t_k}} \quad (3.8)
\]

Where \(\mathbf{v}\), \(W_e\), \(W_d\), \(b_{att}\) are weight-learnable vector, matrix, bias and the softmax( ) function, which is used to normalize \(a^t\) as an attention distribution. The retained encoder hidden vectors are then reweighted by the attention distribution, and compress a hidden vector \(h_t\), known as context vector, which encodes the essential information of document at each time step \(t\):

\[
\hat{h}_t = \sum_{k=1}^{|D|} \hat{a}^t_k h_k \quad (3.9)
\]
In a conventional attentional neural summarization method, the context vector \( \hat{h}_t \) and decoder hidden vector \( d_t \) are fed into a MLP which computes the probability distribution of generating words in a predefined dictionary:

\[
x_{alt} = [\hat{h}_t; d_t]
\]

\[
P_{dict} = \text{softmax}(W_f(\ldots \alpha_1(W_1 x_{alt} + b_1) \ldots ) + b_i)
\] (3.10)

Where \( P_{dict} \) is the probability distribution all over the words in the fixed dictionary, and the probability of the predicted word \( w \) is denoted by \( P_{dict}(w) \). The loss of the target word \( w_t^* \) at time step \( t \) and entire sequence is computed as following:

\[
\text{loss}_t = -\log P(w_t^*)
\]

\[
\text{loss} = \frac{1}{T} \sum_{t=0}^{T} \text{loss}_t
\] (3.11)

The effectiveness of this method has been proven in past work (Rush, et al., 2015). However, this method is not able to produce a word outside a fixed dictionary, and the generated summary usually contains multiple repeated sentences.

Two methods are introduced to address this those drawbacks, one is technique of pointer network and the other is coverage loss function. Pointer network is designed to extract element from input sequence and we leverage this technique to copy words from the document to be summarized:

\[
P_{fixed} = \sigma(W_{\hat{h}t}^t \hat{h}_t + W_{dt}^t d_t + W_{xt}^t x_t + b_{fixed})
\]

\[
P_{final}(w) = P_{fixed} P_{dict}(w) + (1 - P_{fixed}) \sum_{k,w_k = w} \bar{a}_k^i
\] (3.12)

where \( \sigma \) is the sigmoid function; \( x_t \) is the input of decoder, namely the word embedding from previous time step; \( W_{\hat{h}t}^t, W_{dt}^t, W_{xt}^t \) and \( b_{fixed} \) are learnable parameters and bias. \( P_{fixed} \) is a scale between zero and one, which is used to determine output of network should be generated from the fixed dictionary or copied from the original document. Obviously, \( P_{dict}(w) \) is zero if the word is
out-of-vocabulary (OOV) and \( \sum_{k, w_k = w} \hat{a}_k^t \) is zero if \( w \) is not in the source document.

Second, the problem of Repetition occurs in sequence-to-sequence neural model normally. To make model take previous output into concern, coverage vector \( c^t \) is introduced to this framework and equation (3.8) is adjusted as following:

\[
    c_i^t = \sum_{k=0}^t \hat{a}_k
\]

\[
    a_i^t = v^\top \tanh(W_c h_i + W_d d_i + W_c c_i^t + b_{\text{att}})
\]

(3.13)

where \( W_c \) is learnable vector. The changed \( a_i^t \) included the information that have been used makes the model have chance to prevent from generating the repeated component. To make model work as expected, it also is necessary to compute the coverage loss following:

\[
    \text{loss}_t = -\log P(w_i^t) + \lambda \sum_{i} \min (\hat{a}_i^t, c_i^t)
\]

(3.14)

where \( \lambda \) is a hyper parameter to weight coverage loss. Modified loss function encourages model to select those words that are not chosen before.
CHAPTER 4
Experimental Setup

4.1 Speech and language corpora

4.1.1 Corpus for extractive summarization
We conduct a series of experiments on a Mandarin Benchmark broadcast new corpus (Wang et al., 2005). 198 hours of mandarin Chinese TV broadcast news of collected by Academia Sinica and Public Television Service Foundation of Taiwan between November 2001 and April 2003 are included in this corpus. The MATBN dataset is publicly available and has been widely used to evaluate several natural language processing (NLP)-related tasks, including speech recognition (Chien, 2015), information retrieval and summarization (Huang & Wu, 2007; Liu, et al., 2015). As such, we follow the experimental setting used by some previous studies for speech summarization. The dataset has been segmented into separate stories and transcribed manually. Each story contains the speech of one studio anchor, as well as several field reporters and interviewees, of which all sentence boundaries also have been determined by human annotation. A subset of 205 broadcast news documents compiled between November 2001 and August 2002 was reserved for the summarization experiments. Furthermore, since broadcast news stories often follow a relatively regular structure as compared to other speech materials like conversations, the positional information would play an important role in extractive summarization of broadcast news stories. We hence chose 20 documents, for which the generation of reference summaries is less correlated with the positional information (or the position of sentences) as the held-out test set to evaluate the general performance of the
presented summarization framework, while the other subset of 185 documents the training set alongside their respective human-annotated summaries for determining the parameters of the various supervised summarization methods compared in the paper.

On the other hand, twenty-five hours of gender-balanced speech from the remaining speech data were used to train the acoustic models for speech recognition. Table 4.1 shows some basic statistics about the spoken documents of the training and evaluation sets, where the average word error rate (WER) obtained for the spoken documents was about 38.1%.

In addition, a large number of text news documents collected by Central News Agency between 1991 and 2002 (the Chinese Gigaword Corpus released by LDC) are used to train the predefined word embeddings.

4.1.2 Corpus for abstractive summarization
Different from extractive summarization, we use another English corpus, CNN/Daily mail corpus, which contains multi-sentence summary in each document to be
summarized for abstractive summarizer (Nallapati et al., 2016). Since abstractive neural summarizer needs a large training data set to achieve a stable result. This corpus contains 286,226 training pairs, 13,368 developing pairs and 11,487 testing pairs. Each summary, the corresponding reference of per document is written by different authors. The average number of tokens per document and summary are 781 tokens and 56 tokens (or 3.75 sentences). Two versions are provided in the data set, one consisting original entity names and the other replacing those tokens to an entity-id (for example: @entity1: CNN). Here we use the latter and the size of dictionary is 50k.

4.2 Evaluation metrics

Three subjects were asked to create summaries of the documents used in the summarization experiments as the reference (the gold standard) for evaluation. The reference summaries were generated by ranking the sentences in the manual transcript of a spoken document by importance without assigning a score to each sentence. For the assessment of summarization performance, we adopted the widely-used ROUGE measure (Lin C.-Y., 2003). It evaluates the quality of the summarization by counting the number of overlapping units, such as N-grams, longest common subsequences or skip-bigram, between an automatic summary and a set of reference summaries. Three variants of the ROUGE measure were used to quantify the utility of the proposed methods. They are, respectively, the ROUGE-1 (unigram) measure, the ROUGE-2 (bigram) measure and the ROUGE-L (longest common subsequence) measure.
In experiments of extractive summarization, the summarization ratio, defined as the ratio of the number of words in the automatic (or manual) summary to that in the reference transcript of a spoken document, was set to 10% in this research. On the other hand, in the experiments of abstractive summarization, the summary length is constrained in 100 tokens. Since increasing the summary length tends to increase the chance of getting higher scores in the recall rate of the various ROUGE measures and might not always select the right number of informative words in the automatic summary as compared to the reference summary, all the experimental results reported hereafter are obtained by calculating the F-scores of the three variants of the ROUGE measure.

Table 4.2 shows the levels of agreement (the Kappa statistic and ROUGE measures) between the three subjects for important sentence ranking of MATBN corpus. Each of these values was obtained by using the summary created by one of the three subjects as the reference summary, in turn for each subject, while those of the other two subjects as the test summaries, and then taking their average. These observations seem to reflect the fact that people may not always agree with each other in selecting the summary sentences for representing a given document (Liu, et al., 2015).

### 4.3 Features

Since the feature of the abstractive summarizer in this thesis, namely word embedding is learned during training. Here we only introduce features for extractive summarizer.
We exposit two kinds of word embedding features that are Skip-gram (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013) and GloVe (Pennington, Socher, & Manning, 2014), and then compare their differences. Besides, we introduce the traditional features used for spoken document summarization.

4.3.1 Word embedding features

Word embedding is a kind of representation learning, which tries to capture relationships of words from their co-occurrence. An intuitive way to represent words in a numeric way is one-hot representation. However, this method does not provide analogy, since it makes the relation of words are absolutely irrelevant. Therefore, representation learning methods for words are proposed to address this drawback. In this thesis, we leverage two well-known methods, Skip-gram and GloVe. Both of them encode words into vector forms in a cosine space, and have a good property to measure analogy of words. For example, the vector of “queen” minus the vector of “king” is approximately equal to the vector of “woman” minus the vector of “man” (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013). Skip-gram is a shallow MLP whose objective is to find word representations that are effective for predicting the surrounding words. More formally, given a word sequence, $w_{1\sim P}$, this model
maximizes the log probability as following:

\[
\frac{1}{T} \sum_{t=1}^{T} \sum_{c \leq j \leq C, j \neq 0} \log p(w_{t+j}|w_t)
\]  \hspace{1cm} (4.1)

where \( C \) is the context windows, which constrains how many words can influence the central word \( w_t \). Skip-gram learns representation from local information. Similar to Skip-gram, word representations of GloVe are learned from co-occurrence in a constrained context window and the objective function of GloVe is different:

\[
\sum_{i,j=1}^{V} f(X_{ij})(e_i^T e_j + b_i + \bar{b}_j - \log X_{ij})^2
\]

\[
f(x) = \begin{cases} (x/x_{\text{max}})^2, & \text{if } x < x_{\text{max}} \\ 1, & \text{otherwise} \end{cases}
\]  \hspace{1cm} (4.2)

where \( V \) is size of the vocabulary; \( X_{ij} \) is the co-occurrence of \( w_i \) and \( w_j \); \( b_i \) and \( \bar{b}_j \) are biases; \( e_i \) and \( e_j \) are learned word vectors. Since different objective function of both methods result in different relation of words, here we further do some observations.

First, both methods measure relation between words dependent on their cosine similarity. We compute the similarity of a given word with its top-\( N \) similar words. We average cosine similarities of top-\( N \) words, and \( N \) is range from 10 to 200. The result is shown in Figure 4.1, that words in Skip-gram stay more tightly than GloVe.
Second, we calculate top-$N$ overlapping words of a given word in each embedding method and the result is depicted in Figure 4.2. Although both features are driven from the same source (CNA), the overlapping is small. The relevant words of a given vocabulary are very different. For example, there are only 1.34 words that are overlapped in top-10 case. In other word, two methods capture the meaning of a word in different view of aspects, and result in the heterogeneous representations of the same word.

Finally, we explore the relation between word frequencies and top-$N$ words overlapping of both methods, where $N=20$. The result graphically depicted in the Figure 4.3. Word frequencies are rank from high to low in the X axis of Figure 4.3 (left side is higher, and right side is lower), and the number of the X axis is merely the index of words. As shown in Figure 4.3, the higher the frequency the more overlapping words, which means those high-appeared words are tend to be a similar representation in both embedding methods.

4.3.2 Traditional features
Several kinds of indicative features have been designed and widely-used in speech summarization, especially with the supervised summarization methods (Liu, et al.,
The state-of-the-art supervised summarizers compared in this paper (i.e., SVM, Ranking SVM, GCLM and DNN), unless otherwise stated, all use a set of 35 indicative features (as illustrated in Table 4.3) to characterize a spoken sentence, including the lexical features, the acoustic/prosodic features and the relevance features. Among them, the acoustic features were extracted from the spoken documents using the Praat toolkit (Boersma, 2001). We detail three kinds of features as following:

1. Acoustic features

   • Pitch

   A speaker will emphasize the important part of the speech by promoting a higher speaking pitch so as to attract the attention, and low down the pitch when it is not very important. Therefore, pitch can be regarded as a proper acoustic feature for summarization.

   • Energy

   Energy represents the speaker's volume, and is often seen as an important information. In general, when a person emphasizes a thing, the speaker will

<table>
<thead>
<tr>
<th>Table 4.3. Typical features used to characterize spoken documents and their constituent sentences.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Acoustic Features</strong></td>
</tr>
<tr>
<td>1. pitch value (min, max, diff, avg.)</td>
</tr>
<tr>
<td>2. peak normalized cross-correlation of pitch (min, max, diff, avg.)</td>
</tr>
<tr>
<td>3. energy value (min, max, diff, avg.)</td>
</tr>
<tr>
<td>4. duration value (min, max, diff, avg.)</td>
</tr>
<tr>
<td>5. 1&lt;sup&gt;st&lt;/sup&gt; formant value (min, max, diff, avg.)</td>
</tr>
<tr>
<td>6. 2&lt;sup&gt;nd&lt;/sup&gt; formant value (min, max, diff, avg.)</td>
</tr>
<tr>
<td>7. 3&lt;sup&gt;rd&lt;/sup&gt; formant value (min, max, diff, avg.)</td>
</tr>
<tr>
<td><strong>Lexical Features</strong></td>
</tr>
<tr>
<td>1. Number of named entities</td>
</tr>
<tr>
<td>2. Number of stop words</td>
</tr>
<tr>
<td>3. Bigram language model scores</td>
</tr>
<tr>
<td>4. Normalized bigram scores</td>
</tr>
<tr>
<td><strong>Relevance Features</strong></td>
</tr>
<tr>
<td>1. VSM score</td>
</tr>
<tr>
<td>2. LSA score</td>
</tr>
<tr>
<td>3. MRW score</td>
</tr>
</tbody>
</table>
pull the volume louder so as to attract listeners' attention.

- **Duration**

This feature is equivalent to the number of words in the sentence. The more number of words, the more information might be contained in the sentence.

- **Peak and Formant**

Formant is the spectral peaks of the sound spectrum, or the human vocal tract. If the speaker says a sentence with clear articulation, the formant will therefore become higher. On the other hand, if the sentence sounds vague, it may be a non-important part of sentence, and the formant will become lower.

2. **Lexical features**

   - **Bigram Language Model Score**

     The $n$-gram Language Model is a commonly used method in natural language processing, where $n$ means the $n$ previous word sequence. This language model is optimized by maximizing the similarity. The importance of a sentence is calculated by the conditional probability of the word that appears in the statement.

   - **Normalized Bigram Language Model Score**

     Similar to the feature above, this language model score, however takes the length of sentence into consideration. Because the longer the sentence, the higher the more information is contained. However, summary is constrained in a small length, it cannot allow every selected sentence contains too many words.

   - **Named Entities**

     Named entity means the company name, location, name and time and so on.
When discussing a specific domain, the number of named entity will be increased and might have a higher probability to be the important part of the document.

3. Relevance features

These features are driven from different summarization models, mostly based on unsupervised methods. Such as statistical value-based vector space model, graphically based Markov random diffuse and language model based probability generation model (LM).

Also noteworthy is that, for each kind of acoustic/prosodic features, the minimum, maximum, difference and mean value of a spoken sentence are extracted. The difference value is defined as the difference between the minimum and maximum values of the spoken sentence. All the 35 features are further normalized to zero mean and unit variance:

\[
\hat{x}_m = \frac{x_m - \mu_m}{\sigma_m}
\]  

(4.3)

where \(\mu_m\) and \(\sigma_m\) are, respectively, the mean and standard deviation of a feature \(x_m\) estimated from the training set.
CHAPTER 5
Experimental Results

5.1 Baseline experiment for extractive summarization

5.1.1 classic methods with traditional features
At the outset, we evaluate the performance levels of the various supervised summarizers compared in this thesis, i.e., SVM, Ranking SVM, GCLM and DNN. It is worth mentioning that DNN has three hidden layers, while the number of neurons in each layer was determined based on the training set. Note also that all these summarizers are learned from the spoken documents of the training set along with their respective reference summaries, and then tested on the spoken documents of the evaluation set. The corresponding results of these four summarizers (in terms of ROUGE-1, ROUGE-2 and ROUGE-L metrics) are shown in Table 5.1, where TD denotes the results obtained based on the manual transcripts of spoken documents and SD denotes the results using the speech recognition transcripts that may contain speech recognition errors. Furthermore, the results obtained by two other state-of-the-art unsupervised summarizers (i.e., the integer linear programming (ILP) method and the submodularity-based method (Submodularity)) are also listed in Table 5.1 for reference.

Inspection of Table 5.1 reveals three noteworthy points. First, for the TD case, all the four supervised summarization methods tend to perform on par with one another when evaluated with the various ROUGE metrics, although the training objective functions of Ranking SVM and GCLM explicitly take into account the relatedness among summery and non-summary sentences. Second, there is a sizable performance
gap between the TD and SD cases for all the above methods, indicating room for further improvements. We may seek remedies, such as multi-level indexing techniques (beyond merely using words as index terms), to compensate for imperfect speech recognition. We leave this issue for future work. Third, it comes as no surprise that the two celebrated unsupervised summarizers (ILP and Submodularity) are worse than the four supervised summarizers.

5.1.2 Point-wise methods combined with word embedding feature

Because the models we proposed are point-wise methods, here we experiment with the point-wise supervised methods, SVM and DNN with word embedding feature. The word embedding features are driven from the Skip-gram method. The results are shown in Table 5.2. We propose two different models, in model I only the averaged sum of word embeddings are included while in model II we also include the 35 indicatice
features. First, using word embedding feature only are not desirable for both of them.

When combined with traditional features both methods can achieve distinct results in TD cases. However, compared to the Table 5.1, the performance of SVN-II decreases in SD case, which means the word embedding feature misleads the SVM model and results in a worse performance. On the other hand, DNN-II achieves the highest ROUGE scores of itself, which means this model is able to handle data corrupted by ASR errors.

5.2 Experiments on the proposed neural summarizers for extractive summarization

5.2.1 CNN based summarizer

We now turn to the evaluation of the effectiveness of the presented CNN based summarization framework proposed in chapter 3.1, which have two different instantiations. The first instantiation (denoted by CNN-I) takes only the two embeddings of a document-sentence pair and their similarity measure as the input to a
multilayer perceptron (MLP) to induce a ranking score for each sentence, while the second one (denoted by CNN-II) additionally includes the 35 indicative features (which are also used by the state-of-the-art supervised summarization methods compared in this paper) as part of the input of MLP. Both CNN-I and CNN-II employ different sets of 50 filters, respectively, for their two individual convolutional layers (thereby leading to 50 feature maps for CNN-I and CNN-II, respectively), where all filters have a common size of 5 consecutive words. Furthermore, the pooling layers of CNN-I and CNN-II all adopt the max pooling operation.

The corresponding results of these two methods are shown in Table 5.3; a closer look at these results reveals two things. First, for the TD case, both CNN-I and CNN-II outperform the supervised or unsupervised summarization methods compared in this paper by a significant margin. Furthermore, CNN-II is superior to CNN-I, indicating the benefit of including extra indicative features for speech summarization. Second, for the SD case, the performance gains offered by CNN-I and CNN-II are diminished. CNN-I and CNN-II seem to perform comparably to, or slightly worse than, the existing state-of-the-art supervised summarization methods. The reasons for this phenomenon, however, await further in-depth studies.

### 5.2.2 CNN-LSTM based summarizer

At the outset, we evaluate the effectiveness of the presented CNN-LSTM based summarization framework. Besides, in order to find a better representation of word
embedding in the SD case, which the performances decrease with a distinct margin when compared with TD cases, we leverage various representative word embedding methods, including the skip-gram model, the global vector model (GloVe) and their combination. Similar to the previous CNN models, we also exam the capabilities of the handcraft features used above, two instantiations are included. On one hand, the first instantiation (denoted by CNN-LSTM-I) takes only the two embeddings of a document-sentence pair and their similarity measure as the input to a multilayer perceptron (MLP) to induce a ranking score for each sentence. Opposite to CNN-LSTM-I, the second one (denoted by CNN-LSTM-II) additionally includes the

<table>
<thead>
<tr>
<th>TD</th>
<th>CNN-LSTM-I</th>
<th>ROUGE-1</th>
<th>ROUGE-2</th>
<th>ROUGE-L</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Skip-Gram</td>
<td>0.493</td>
<td>0.390</td>
<td>0.449</td>
</tr>
<tr>
<td></td>
<td>GloVe</td>
<td>0.459</td>
<td>0.346</td>
<td>0.413</td>
</tr>
<tr>
<td></td>
<td>Skip-Gram + GloVe</td>
<td>0.485</td>
<td>0.382</td>
<td>0.439</td>
</tr>
<tr>
<td>SD</td>
<td>Skip-Gram</td>
<td>0.370</td>
<td>0.217</td>
<td>0.324</td>
</tr>
<tr>
<td></td>
<td>GloVe</td>
<td>0.332</td>
<td>0.186</td>
<td>0.288</td>
</tr>
<tr>
<td></td>
<td>Skip-Gram + GloVe</td>
<td>0.331</td>
<td>0.183</td>
<td>0.280</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SD</th>
<th>CNN-LSTM-II</th>
<th>ROUGE-1</th>
<th>ROUGE-2</th>
<th>ROUGE-L</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Skip-Gram</td>
<td>0.520</td>
<td>0.420</td>
<td>0.472</td>
</tr>
<tr>
<td></td>
<td>GloVe</td>
<td>0.488</td>
<td>0.387</td>
<td>0.449</td>
</tr>
<tr>
<td></td>
<td>Skip-Gram + GloVe</td>
<td>0.525</td>
<td>0.430</td>
<td>0.485</td>
</tr>
<tr>
<td></td>
<td>Skip-Gram</td>
<td>0.385</td>
<td>0.253</td>
<td>0.346</td>
</tr>
<tr>
<td></td>
<td>GloVe</td>
<td>0.391</td>
<td>0.255</td>
<td>0.349</td>
</tr>
<tr>
<td></td>
<td>Skip-Gram + GloVe</td>
<td>0.404</td>
<td>0.268</td>
<td>0.364</td>
</tr>
</tbody>
</table>

Table 5.4. Summarization results achieved by leveraging different word embedding methods in the CNN-LSTM summarization framework.

Table 5.5. Summarization results achieved by further incorporating the typical features in the CNN-LSTM summarization framework.
35 indicative features as part of the input of MLP. Both CNN-LSTM-I and CNN-LSTM-II employ different sets of filters according to performances, respectively, for their two individual convolutional layers (thereby leading to numbers of feature maps for CNN-LSTM-I and CNN-LSTM-II, respectively), where all filters have a common size of 5 consecutive words. Furthermore, the pooling layers of CNN-LSTM-I and CNN-LSTM-II all adopt the max pooling operation. The corresponding results of these four summarizers are shown in Tables 5.4 and 5.5, the same as the previous experiments, where TD denotes the results obtained based on the manual transcripts of spoken documents and SD denotes the results using the speech recognition transcripts that may contain speech recognition errors.

A closer look at these results reveal several notable things. First, the skip-gram model consistently outperforms the glove model in most cases. To our surprise, the combination of both word embedding methods doesn’t lead to the best improved results when paired with CNN-LSTM-I for both TD and SD cases, whereas it is the best system when being paired with CNN-LSTM-II. A possible reason might be the different strength of relevance between words of Skip-gram and GloVe. The former provides strong connections between words and therefore provides more information to the model when only using Skip-gram feature, the latter products weaker tightness of words and probably is not enough to the model. However, since the tightness of words embedding features provided by GloVe are not strong, this feature might be easier to combine with other features. The phenomena can be seen on the CNN-LSTM-I & II. Compared with Skip-gram, CNN-LSTM based models with GloVe feature benefit from indicative features more than Skip-gram. Moreover, CNN-LSTM-I with GloVe is worse than the one using Skip-gram in both TD and SD cases, but CNN-LSTM-II model with GloVe becomes the better one in SD case. Yet
the reasons are worthy to be examined thoroughly in our future work. Second, when comparing Tables 5.4 and 5.5, the results reveal that the indicative features can provide extra important clues to complement the data-driven embeddings learned by the neural networks. The results also encourage us to learn representative acoustic clues from speech signals directly in the future to enhance speech summarization performance.

Finally, we compare this summarizer with the classic and CNN-based methods. According the results above, that Skip-gram provides the model better summarization ability than GloVe most of time, we only compare CNN-LSTM with CNN based framework using Skip-gram feature and combined with traditional features. Although the results show CNN-LSTM-based methods using Skip-gram word embeddings cannot achieve better results or even worse in TD cases when compared CNN-based framework, in SD cases, CNN-LSTM-I shows a clear improvement to CNN-I, and is comparable to the best model, Ranking SVM with a tiny distinction. Furthermore, CNN-LSTM-II with words embedding driven from GolVe only is already better than Ranking SVM, and outperform all the methods when combining with Skip-gram features and traditional features. Since CNN-LSTM framework seem to capture more useful information so that it performs better results in SD cases, which contain a lot of corruptions caused by ASR errors, a possible reason makes performances of this

<table>
<thead>
<tr>
<th></th>
<th>ROUGE-1</th>
<th>ROUGE-2</th>
<th>ROUGE-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attention</td>
<td>0.293</td>
<td>0.105</td>
<td>0.268</td>
</tr>
<tr>
<td>Attention + pointer</td>
<td>0.340</td>
<td>0.138</td>
<td>0.303</td>
</tr>
<tr>
<td>Attention + pointer + coverage</td>
<td>0.389</td>
<td>0.167</td>
<td>0.355</td>
</tr>
</tbody>
</table>
model decreased in TD cases might be the model keeping too much information and results in worse performances. Although the reasons still wait to be clarified, the results show this framework has ability to handle the corrupted information more delicately.

5.2.3 Abstractive neural summarizer

The input length of the abstractive neural summarizer is limited to 400 tokens and output length to 35~100 tokens. The results are shown in Table 5.6, and a comparison of different abstractive summarization models are provided in Figure 5.1. Note that “attention + pointer” summarizer is the framework combined with pointer network, and “attention + pointer + coverage” means the model is optimized by the coverage loss in the end of training phase. Compared to the conventional attention neural model similar to (Rush, et al., 2015), the attention + pointer summarizer products a much better result and abstractive summarizer with coverage loss outperforms both of them.

Figure 5.1 Comparison of abstractive summarization models with ground truth
Those results indicate that addressing the problems of OOV and repeating problem are an effective way to improve the performance of an abstractive summarizer.
CHAPTER 6
Conclusion and Future Work

In this thesis, we have presented CNN-based and CNN-LSTM-based supervised summarization methods for use in speech summarization, which were empirically compared to several well-developed summarization methods. The CNN-based methods achieve the best results when compared with other well-known methods, and the CNN-LSTM-based methods seem to handle spoken documents well, which are corrupted by ASR errors more delicately. Both method seem to hold promise for further development in the context of text and speech summarization. On the other hand, the abstractive neural summarization methods seem to produce a rewritten summary in an effective way.

As to future work, we envisage several directions for extractive summarization. First, we plan to employ disparate learning to rank (training) paradigms, such as pair-wise and list-wise learning, for estimating the model parameters of the CNN and CNN-LSTM based methods. Second, we will explore more sophisticated summarization frameworks and their synergies with start-of-the-art summarization methods. Third, we are also interested in investigating robust indexing techniques for representing spoken documents in order to bridge the performance gap between the TD and SD cases.

For abstractive summarization, we will explore the ways to eliminate the difference between training phase and testing phase for abstractive neural summarization methods. The abstractive neural summarization methods in this thesis are trained in maximizing likelihood criteria, whose decoder receives the words from correct summary in training phase, yet in the testing phase the decoder have to depend
on the previous output. Therefore, we will investigate the reinforcement learning for abstractive neural summarizer. We will change the objective function so as to make decoder take the sampled output as input by itself by leveraging reinforcement learning techniques.
Bibliography


Compositionality (CVSC).


Spoken language understanding: Systems for extracting semantic information from speech (pp. 357-396).


sentence summarization. *The SIGNLL Conference on Computational Natural Language Learning (CoNLL)*.


