Master's Thesis in Graduate School of Library, Information and Media Studies

Study on Effect of Semantic Content Generalization to Pointer Generator Network in Text Summarization

March 2021

201921656

WU YIXUAN

Study on Effect of Semantic Content Generalization to Pointer Generator Network in Text Summarization
文書要約におけるポインタジェネレータネットワークに対する意味論的内容一般化の影響に関する研究

Student No.: 201921656
氏名: 呉　宜暄
Name: WU YIXUAN

Semantic content generalization is a method for text summarization that reduces the difficulty of training neural networks by replacing some phrases such as named entities with generalized terms. The semantic content generalization has achieved remarkable results in enhancing the performance of the sequence to sequence attention model. Besides that, the pointer generator network could ease the training of the summarization based on a mechanism that copies words from the original text, which shares a similar idea with semantic content generalization. There are two purposes of this work. Firstly, we test and verify the effect of semantic content generalization on the pointer generator network in text summarization. Secondly, we attempt to find out the effect of semantic content generalization on the pointer generator network when the number of dictionaries increased and whether the AutoNER can improve the performance of the semantic content generalization or not. In this paper, we proposed two methods for semantic content generalization in pre-processing and combine the semantic content generalization with the pointer generator network.

We examined the performance through the experiments using CNN/DailyMail datasets. From the experiment, we found that semantic content generalization can improve the performance of the pointer generator network. We evaluated our model with ROUGE metrics, which measures the similarity between summaries. With the pre-processing using the Named Entities-driven Generalization and dictionaries, the scores of the ROUGE-1 can be improved by 0.0037 in the best settings. However, we regret finding that AutoNER cannot effectively improve the performance of semantic content generalization, and analyzed the reason why AutoNER cannot work well in the experiments. The experimental results imply that post-processing is the main reason which makes many errors in summarization and results in the performance lower than the expectation.

Principal Academic Advisor: Kei WAKABAYASHI
Secondary Academic Advisor: Yohei SEKI

Study on Effect of Semantic Content Generalization to Pointer Generator Network in Text Summarization

WU YIXUAN

Graduate School of Library,
Information and Media Studies

University of Tsukuba

March 2021

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, with the explosive growth of text information, people can access a large amount of text information every day, such as the news, blogs, reports, paper, and so on. However, the problem of information overload also followed. And we need to filter out the most useful information from a large amount of text information every day. Therefore, how to select the part that is beneficial to a user from the massive amount of information becomes particularly important. And automatic text summarization is one of the key technologies for helping the selection process [2].

According to Radev's definition, the summarization is "A text that is produced from one or more texts, that conveys important information in the original text(s) and that is no longer than half of the original text(s) and usually significantly less than that." [3, 4] Therefore, according to this definition, the purpose of the automatic text summarization technology is to automatically output short and very general text through the machine.

The automatic text summarization can be applied to a wide range of scenarios. It is not only the generalization of the text in the traditional sense, but also through the automatic text summarization to title the text and even can automatically generate the reports.

Although the application scenarios of automatic text summarization are very large and wide, the development of this field is actually insufficient, because computers are different from the human brain. For computers, generating summaries is a very challenging thing. Even if just choose and splice from the original text, it needs to have a certain understanding of the text to determine whether the extracted sentence is an important sentence, and then generate a natural short text. Therefore, automatic text summarization is often combined with related theories of natural language processing. Especially with the development of deep learning in recent years, automatic text summarization technology has made certain progress.

Automatic text summarization can be categorized into two types: extractive summarization and abstractive summarization. Extractive summarization is a method that judges the important sentences in the original text and extracts these sentences as a summary. Abstractive summarization uses the advanced natural language processing algorithm to generate a more concise summary through the technology of paraphrase, synonym replacement, sentence abbreviation, and so on. Although abstractive summarization needs to understand the content of the article, and this will be a complicated process, the development of deep learning has made the generation of summaries feasible in recent years [5].

The sequence to sequence model was first proposed by Sutskever et al. [6] in 2014 and was

applied in the field of machine translation. Due to the flexibility of the sequence to sequence model, it has also been used in the field of the automatic text summarization [1, 7–10]. However, since the summarization tasks usually require us to transform a long sentence, the summary generation by the sequence to sequence models tends to be unstable, e.g., generating irrelevant frequent words and repeating the same words many times.

A neural network model called pointer generator network proposed by See et al. [1] addresses these issues of the sequence to sequence models. The pointer generator network incorporates a function that copies words from the source text and a mechanism that avoids the repetition of the same words, which largely mitigates the issues in the sequence to sequence models. This can be regarded as one of the combinations of the extractive and abstractive approach to attempt to take both advantages, which extract important parts and generate new texts.

In this paper, we focus on the extraction of semantically coherent phrases in the abstractive summarization. For example, when the important part of an original text includes the name of a person, the summary should contain the name of the person as it is without any editing. Compared to the sequence to sequence models, the pointer generator networks probably can extract and generate semantically coherent phrases more easily due to the mechanism of copying. However, the extraction performance of the pointer generator network from this viewpoint has not been verified.

Related to this direction, Kouris et al. [9] proposed a method called the semantic content generalization in 2019. The semantic content generalization is a method that replaces a semantically coherent phrase with a special token (e.g., replacing "Martin Luther King Jr." with "_person_") as a pre-processing for the training and prediction of the neural networks. This pre-processing reduces the complexity of sequence transformations so that the neural networks do not need to memorize subsequences (phrases) that indicate the semantic contents. Kouris et al. [9] applied the semantic content generalization to the sequence to sequence model and empirically showed the effectiveness of the method, but its utility on the pointer generator networks is still unclear.

Since the pointer generator network and the semantic content generalization seem to share a similar idea, we need to clarify the relationship between these two independent but combinable methods. In this paper, we examine whether the semantic content generalization can improve the pointer generator network or not. From the experiment, we draw some insights on the property of the pointer generator networks, which will be either of the possible conclusions below:

1. If the combined method improves the summarization performance, it implies that the semantic content generalization can help the pointer generator network to extract semantically coherent phrases, and at the same time, it also implies that the pointer generator networks were struggling to learn the extraction of semantically coherent phrases sufficiently.

2. If the combined method does not improve the performance, it implies that the pointer generator network has a sufficient capability to learn the extraction of semantically coherent phrases easily.

The experimental result in this paper implies that the former conclusion likely to be supported. In the following chapters, we present the detail of the methods and the processes

that are employed in the experiment.

Furthermore, we have created a dictionary for semantic content generalization. Assumed that the semantic content generalization can improve the pointer generator, and until now that the semantic content generalization achieved by using the Stanford NER (NER is abbreviated of Named Entity Recognition) which has only 3 categories. Therefore, we created multiple categories of dictionaries for semantic content generalization and try to find out the relationship between the quantity of the dictionary and the performance improvement of the pointer generator network.

In 2018, Shang et al. [11, 12] proposed AutoNER model which can do the named entity recognition automatically and only with the specific domain dictionaries. Considering that the achievement of the AutoNER, we tried to combine the pointer generator network with the AutoNER which will be used as the pre-processing. And verify whether the AutoNER can improve the effectiveness of the semantic content generalization.

The contribution of this thesis is summarized as below:

- We examine the effect of the semantic content generalization method on the pointer generator network through two different pre-processing.
- We propose new methods for semantic content generalization which are called dictionary-based generalization and AutoNER-based generalization. These two methods can deal with multiple categories and the most can reach 9 different categories.

For the rest of this paper, we will introduce the related work in chapter 2. And introduce the method which had been proposed in this paper in chapter 3. The result of the experiment and the relevant analysis will be described in chapter 4. Finally, chapter 5 and chapter 6 will describe the discussion and the conclusion.

Chapter 2

Related Work

With the development of deep learning, more and more models and networks were proposed by the researchers. Among these models, the sequence to sequence models is the most used model, and the pointer generator networks improve the sequence to sequence model. And Kouris et al. proposed an innovative method for pre-processing in 2019. We will introduce these models and networks below.

2.1 Sequence to Sequence Model

In 2014, Sutskever et al. [6] proposed a method that used the end to end neural network model. The end to end neural network model can complete the map sequences to sequences tasks. The author used one LSTM as the encoder for encoding, and another LSTM as the decoder for decoding. And it was applied to the English-French machine translation task, even almost achieved the best score at that time. The author found two interesting points.

1. The model is not sensitive to the active voice and the passive voice, but sensitive to the order of the inputted words.
2. The input sentences in reverse order can improve the model performance.

For the field of text summarization, although the use of research methods in other fields can achieve unexpected results, the sequence to sequence model proposed by Sutskever et al. is mainly for machine translation, which is different from text summarization. For the task of text summarization, if some text features are added to the model, it may be of great help to the performance of the model. Therefore, in 2016, Nallapati et al. [7] improved the performance of the abstract summarization model through multiple improvements based on the encoder-decoder structure and achieved better performance through training and testing based on the English corpus. They improved the model from the following four aspects:

1. Introduce more external language feature information.
2. Use the hierarchical attention mechanism to extract word-level and sentence-level attention information and then multiply them accordingly.
3. A simple pointing judgment mechanism that can generate and copy the original text to reduce the appearance of unknown words and low-frequency words.

4. Solve the problem of excessive prediction vocabulary of the decoder through negative sampling [13].

2.2 Pointer Generator Network

In 2017, See et al. [1] proposed a summary model based on the idea of the pointer generator. In terms of model architecture, it is an improved model based on the sequence to sequence model. The authors of this paper propose that only using the sequence-to-sequence model to generate abstracts will bring the following two problems:

1. Unable to accurately copy details, and may not be able to deal with out-of-vocabulary words (OOV).

2. The possibility of repetition.

Therefore, the pointer generator network has been improved in two ways. Firstly, the network can copy words from the source text through pointers, which helps to accurately copy information. At the same time, it can ensure the ability to generate new words through the generator. The key point of the pointer generator network is the introduction of the extra variable, and through this variable to characterize the probability of generating the word at the current time step from the vocabulary. Meanwhile, after some calculating, we can also get the probability of copying the word at the current time step. Secondly, the method uses the coverage mechanism to track the summarized content to prevent duplication. Specifically, this method solves the problem of unknown words and low-frequency words by introducing weights, combining the generation probability and copy probability of summarized vocabulary, and referring to the idea of solving the problem of "over translation" and "under translation" in machine translation [14]. At the same time, since the objective function of generating the summarization takes into account the attention weight of certain words which is decoded at the decoder side to reduce the possibility of generating the same word, the problem of repeated words in the summarization can be reduced to some extent.

In summary, the innovation of the pointer generator networks has two aspects.

1. Use the hybrid pointer-generator network which can copy text by the pointers and generate the words by the generator.

2. Use the coverage mechanism which used the coverage rate to track the summarized parts.

The pointer generator network was tested on the English datasets and has better performance than the summary model proposed by Nallapati et al. [7]. And about the details of the pointer generator network, such as the architecture of the network and the calculation, we will describe it as the section of the method in chapter 3.

2.3 Semantic Content Generalization

In 2019, Kouris et al. [9] proposed a semantic content generalization method on abstractive text summarization based on the sequence to sequence attention model. The method

enhances the performance of abstractive text summarization based on the sequence to sequence model by replacing phrases with generalized terms. Because the named entity is a kind of semantic content, they proposed two approaches for generalization the one of it called named-entity generalization (NEG) and another one is called level-based generalization (LG).

NEG generalizes only those named-entities that are detected by named entity recognizer trained to detect entities of location, person, and organization [15–17]. LG uses the concept of generalization to preprocess based on a dictionary such as WordNet [18, 19]. It is mainly to replace low-frequency words with high-frequency words.

The authors make seven definitions of semantic content generalization. Respectively are the taxonomy of concepts, hypernym, the taxonomy path of concept, the taxonomy depth of concept, the generalization concept, the generalization text, and the level of generalization.

The taxonomy of concepts can be easier to understanding. It means that the concepts can be extracted from the source text. For the hypernym, the author has built a hierarchical structure. For example, the hypernym of the banana is the fruit. The taxonomy path of concept means that the path where the concept has been extracted. And the extracted concepts are the ordered sequence, and the numbers of the concepts determine the taxonomy depth of the concept. The concept can be generalized only when the concept contains a generalizable concept. After generalizing the concept, we will obtain the generalizable text. It means that the generalizable text contains at least one generalizable concept which has been replaced by the superordinate concept. And as the author described, the minimum taxonomy depth of a generalized concept constitutes the level of generalization of the given text.

The author uses the sequence to sequence model as the summarizer and uses the method which is mentioned above to do the pre-processing. And then, after obtaining the intermediate summarization, it is not finished, the final output still can not be gotten. Because the intermediate summarization needs to do the post-processing. The example of the taxonomy of five concepts is shown in Figure 2.1.

The datasets adopted by the Kouris et al. to evaluate the method were Gigaword and DUC2004. And we will describe the detail of the process in chapter 3.

2.4 AutoNER

In 2018, Shang et al. [11, 12] proposed a model that can learn the named entity tagger using the domain-specific dictionary.

With the development of deep learning, we can train the network only when we have enough data at the most time. Especially, some specific tasks, but sometimes we can not obtain a great deal of data. To solve the problem, Shang et al. proposed a model named AutoNER. Most of the named entity recognition is based on a large amount of the labeled data. However, it is difficult to have numerous labeled datasets, and AutoNER is a method that deals with the situation when we do not have enough datasets. Shang et al.tried to solve the problem by following two methods.

1. To find the labeled datasets of a similar field.

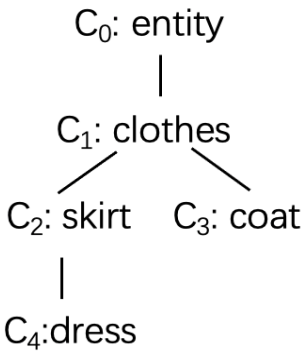

Figure 2.1: A taxonomy of concepts.

2. Use the idea of distant supervision which means that use the domain-specific dictionary to generate some labeled data.

The authors pointed these two points to us, and of course, still exist other methods to solve the problem.

About the distant supervision which Shang et al. mentioned in their paper, there is an example to explain it. For example, we have a dictionary-like that:

Table 2.1: The example of the dictionary

country	*company*	*person*
Japan	Mitsubishi Group	Iwasaki Yataro
China	Baidu	Robin Li
America	Apple	Steve Jobs
Korea	Samsung	Lee Byung-chul
Germany	SIEMENS AG	Ernst Werner von Siemens

The dictionary just as Table 2.1 shows, and we can use the dictionary to label the sentence. For the sentence like, "Baidu is a Chinese company and its founder is Robin Li.", we can label it as the following.

"<company>Baidu</company> is a Chinese company and its founder is <person>Robin Li</person>."

After that, we can obtain the sentence which has been labeled. However, without a doubt, we can not use the labeled corpus like that directly because of the two reasons below.

1. We can not covered all named entity, as a result of the limit of the dictionary. And the named entity of unknown can consider as the named entity which can apply for all types.

2. In the situation that one named entity has multiple types, we can not know that which one is we need.

Therefore, the author proposed the fuzzy CRF which is a sequence labeling method that can be trained with partially annotated sentences. Although, the fuzzy CRF can deal with

the situation that one named entity has multiple types, still does not solve the problem that the noise has been brought by distant supervision. And then, the authors proposed a method called the tie or break to replace the traditional sequence labeling scheme in order to make the effect of noise on the models as smaller as better.

With the layer of the fuzzy CRF and the scheme of the tie or break, the authors proposed the AutoNER. Shang et al. examined the models with three basic datasets. Among the datasets, there are two in the biomedical domain is BC5CDR and NCBI, and the last one is from the SemEval 2014 challenge named LaptopReview, and the result of the experiment proved that the performance of the AutoNERis superior. Through AutoNER we can do the named entity recognition and without any additional human effort.

Chapter 3

Proposed Method

The summarization model used in the paper that proposed the semantic content generalization [9] is the sequence to sequence model. However, See et al. have already improved the sequence to sequence model and proposed the pointer generator network. Therefore, we used the pointer generator network as the summarizer and expect it can improve the performance of the semantic content generalization. We will introduce the pointer generator network in chapter 3.1.

In this paper, we also attempt to improve the semantic content generalization method by applying multiple category dictionaries and AutoNER method. Semantic content generalization has proved that it can improve the performance of the sequence to the sequence model. And we considered that whether the multiple category's dictionaries can improve the performance of summarizers too. Therefore, we created multiple categories of dictionaries for semantic content generalization and applied them to AutoNER for automatic recognition. And we will introduce these in detail in chapter 3.2.

The overall flow of the method is shown in Figure 3.1. Firstly, the datasets will be processed with the pre-processing method and output datasets which have been replaced by the named entity phrases with a symbol that represents the named entity type. Then, we input the processed datasets into the pointer generator and obtain an intermediate summarization. Finally, through post-processing that restores the replaced symbol with the original words, then we can get the final output. In the following section, we will introduce the pointer generator network, pre-processing, and post-processing.

3.1 Pointer Generator Network

The pointer generator network uses the traditional sequence to sequence network as the underlying base. Since BiLSTM can capture long-distance dependency and position information, the encoder adopts BiLSTM and the decoder is LSTM.

The sequence to sequence model with the attention mechanism that See et al. used is shown in Figure 3.2. And the pointer generator network which they proposed is shown in Figure 3.3.

From Figure 3.2, the words are input to the encoder one by one, and the encoder will produce a sequence of encoder hidden state. And on each step, the decoder will receive the word embedding of the previous word and decoder state. But the previous word when training is different from the previous word at the test time. When training time, the

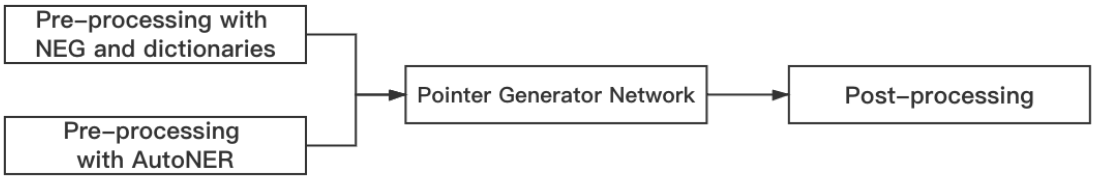

Figure 3.1: Overall flow chart.

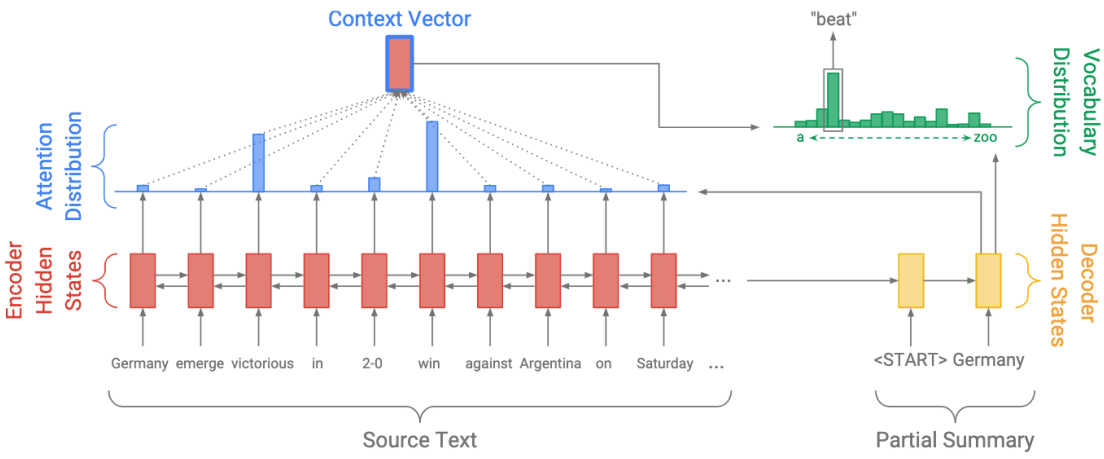

Figure 3.2: Sequence to sequence model with attention mechanism. [1]

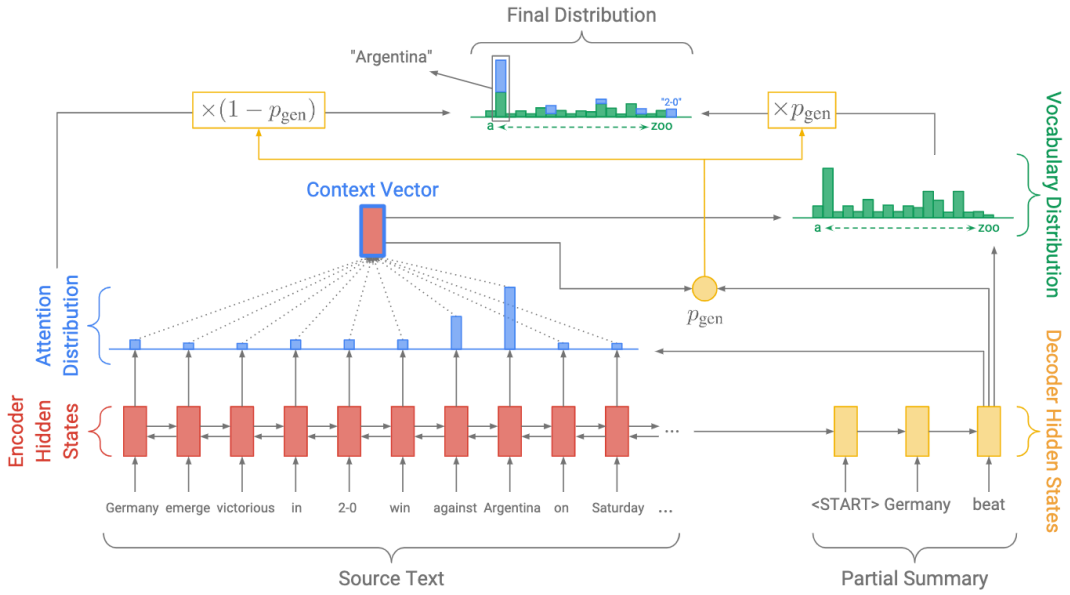

Figure 3.3: Pointer generator network. [1]

previous word represents the word of the reference summary. And at the test time, the previous word represents the word that is emitted by the decoder.

The attention distribution is calculated as in Bahdanau et al. [20], which represent a probability distribution over the source words.

$$e_i^t = v^T tanh(W_h h_i + W_s s_t + b_{attn}) \tag{3.1}$$

$$a^t = softmax(e^t) \tag{3.2}$$

At the same time, the attention distribution can tell the decoder where should pay attention to and produce the next word. And it also used to produce the weighted sum of the encoder hidden states which called the context vector $h_t{}^*$.

$$h_t{}^* = \sum_i a_i{}^t h_i \tag{3.3}$$

The context vector can be understood as an input fixed-size representation for this step. When we combine the context vector with the decoder state and input them to the two linear layers, we can get the vocabulary distribution P_{vocab}.

$$P_{vocab} = softmax(V^{'}(V[s_t, h_t{}^*] + b) + b^{'}) \tag{3.4}$$

The vocabulary distribution is a probability distribution over all words in the vocabulary. And we can obtain the final distribution through the vocabulary distribution.

$$P(w) = P_{vocab}(w) \tag{3.5}$$

The loss of target word for each time step has used the negative log-likelihood.

$$loss_t = -logP_{(w_t^*)} \tag{3.6}$$

And the overall loss for the whole sequence can be calculated by the following formula.

$$loss = \frac{1}{T}\sum_{t=0}^{T} loss_t \tag{3.7}$$

The pointer generator network like a hybrid-model since it combined the sequence to sequence model with the pointer network which has been proposed by Vinyals et al. [21].

And as can be seen from Figure 3.2 and Figure 3.3, the pointer generator network introduces p_{gen} and determines the final distribution together by the vocabulary distribution and attention distribution. p_{gen} is a generation probability ranging from 0 to 1. It can be used to determine the probability of whether to generate a word from the vocabulary or copy a word from the source text.

$$p_{gen} = \sigma(W_{h^*}^T h_{t^*} + W_s^T s_t + W_x^T x_t + b_{ptr}) \tag{3.8}$$

The pointer generator also has an extended vocabulary that is the union of the vocabulary and all words in the source document. Then, we can obtain the following probability distribution like that:

$$P(w) = p_{gen}P_{vocab}(w) + (1 - p_{gen})\sum_{i:w_i=w} a_i^t \tag{3.9}$$

In contrast to the sequence to sequence model, the pointer generator has the following advantages:

1. The introduction of p_{gen} makes it easier to generate words from the original text by pointer generator network than sequence to sequence model based on attention mechanism.

2. The pointer generator network can even copy informal words in the source text. This is also the main advantage brought by the network, so it can generate words that do not appear in the training corpus and can use a smaller vocabulary without sacrificing performance. It also requires less computing resources and storage space.

The authors added a coverage mechanism to the network [14], which solved a problem in sequence-to-sequence model that generates the same word repetitively. The coverage mechanism uses the mechanism of machine translation to solve "over translation" and "under translation". It obtains the coverage vector by summing the attention weights of the previous time step and then influences the current attention weight through the previous attention weight, and the formula is below.

$$c^t = \sum_{t'=0}^{t-1} a^{t'} \tag{3.10}$$

The coverage mechanism mainly maintains the coverage vector, which is the sum of the attention distribution of all previous decoder time steps. c^t is the distribution over the source document words that is calculated using the attention mechanism about the degree

of coverage. And the coverage vector used as the extra input for the attention mechanism, so the formula 3.1 should be changed into:

$$e_i{}^t = v^T tanh(W_h h_i + W_s s_t + w_c c_i{}^t + b_{attn}) \tag{3.11}$$

See et al. defined a coverage loss to penalize repeatedly attending to the same locations:

$$covloss_t = \sum_i min(a_i^t, c_i^t) \tag{3.12}$$

Because the coverage loss is bounded in the field of machine translation, so to make their coverage loss more flexible, See et al. introduced the hyper-parameter λ to the primary loss function and then obtain a new composite loss function.

$$loss_t = -logP_{(w_t^*)} + \lambda \sum_i min(a_i^t, c_i^t) \tag{3.13}$$

3.2 Semantic Content Generalization

3.2.1 Pre-processing

- **Named Entities-driven Generalization**:

  NEG is an abbreviation of named entities-driven generalization. It generalizes only those named entities such as location, person, and organization.

  Kouris et al. proposed NEG which uses the Stanford log-linear part-of-speech tagger [22] to recognize the noun phrases because the NEG only applied for the noun phrases. And using the Stanford NER [17] and WordNet [18, 19] to extract the named entities.

  However, the NEG applied on the pointer generator network is simplified, we only use the Stanford NER to extract the named entities. At the same time, we have not to use the Stanford log-linear part-of-speech tagger to recognize the noun phrase.

  We pass the datasets through the pre-processing of simplified NEG and replace named entities whose taxonomy path contains specific named entities with a special symbol that indicates the entity types. To formulate NEG, the author proposes several concepts.

  1. Taxonomy of concepts: Concept classification consists of a hierarchy of concepts related to is-a relationship types.
  2. Taxonomy path of concepts: Let C_a be a concept. For a given taxonomy of concepts, C_a's taxonomy path P_{C_a} is an ordered sequence of concepts $P_{C_a} = \{C_a, C_{a+1}, ..., C_n\}$, and C_i semantically contains C_j, C_j is the hypernym of C_i. C_n is the root concept of taxonomy.

  When the frequency of terms in the input text is less than the specified threshold θ_f, and it's classification path p_i contains a named entity $c \in E$, it can be generalized. In this case, C_i will be replaced by its superordinate word C. The output is the generalized text (which the original paper [9] refers as "genText") of the input text. When the threshold is equal to infinity, NEG's algorithm is similar to the operation of named entity anonymization proposed by Hassan et al. [23].

Algorithm 1 Pre-processing of Named Entities-driven Generalization

Require: $E = \{\text{Location, Person, Organization}\}$, document d

1: Apply the named entity recognizer to d and store the set of recognized (phrase,entity type) pairs to R
2: **for** each (phrase, entity type) pairs $(w, e) \in R$ **do**
3: Replace w in d with e
4: **end for**

The example of the NEG is shown in Table 1.

Table 3.1: The example of the NEG

Input text	doctor arrived at several locations near Los Angeles after receiving the help message but no wounded people was found.
Generalized text	doctor arrived at several locations near __location__ after receiving the help message but no wounded people was found.
Generalized summary	doctor arrived at several locations near __location__.
Output summary	doctor arrived at several locations near Los Angeles.

We only refer to the idea of the NEG and have not set the threshold like them.

After generalizing the generalizable concepts, input it into the summary model for the summary, and finally, restore the replaced words through post-processing. The specific post-processing method will be explained later.

- **Dictionary-based Generalization**:

  Based on NEG, we create a simple dictionary that contains six categories: vehicles, weather, sports, crime, disease, and career. The parts of entries of vehicles[1], weathers[2], sports[3] and crimes[4] in the simple dictionary are obtained through online searches. At the same time, the entries of disease, career, and the rest of the vehicles, weather, sports, and crime are extracted from the datasets.

  We chose disease, career, vehicles, weather, sports, and crime as the dictionaries because we considered that the frequency of these 6 categories in the datasets is relatively higher than others. It is the dictionaries for the experiment in this paper, and those dictionaries are created according to the datasets. The effect of the different domains between the dictionary and datasets will become one of the future research topics.

  We selected three categories from the dictionary and add it to the concept set(person, location, and organization) as the 6 categories dictionary and add all the dictionaries to the concept set as the 9 categories dictionary. Since the number of phrases and words is still small, we chose the categories with the largest number of phrases and words. And the number of entries in each category is shown in Table 3.2, and the total items in the dictionary are 796. Therefore, we selected the "weather", "crime" and "career" as the 6 categories.

[1] `https://englishstudyonline.org/types-of-vehicles/`
[2] `https://www.enhancedlearning.com/wordlist/weather.shtml\#wls-id-22`
[3] `https://www.lingokids.com/english-for-kids/list-of-sports`
[4] `https://critical.findlaw.com/critical-charges/view-Allcriticalcharges.html`

Table 3.2: The numbers of entries in each categories

weather	*career*	*crime*	*sports*	*vehicles*	*disease*	*total*
282	150	134	131	63	36	796

We identify phrases or words belonging to categories based on dictionary lookups. The example of the dictionary is shown in Table 3.3.

We preprocess CNN/DailyMail and replace the phrases or words belonging to the named entities of "person", "location", "organization", "weather", "sports", "crime", "vehicles", "disease" and "career" in the datasets to their superordinate concept. Then, the labeled datasets are input to the pointer generator network to obtain an intermediate summary.

In the current experimental stage, due to the limited number of phrases and words in the dictionary, generalization replacement is performed on all phrases or words in the category dictionary, and the replacement threshold has not been set for the phrases or words which is needed to be replaced. In future experiments, to explore whether the threshold will affect the summary results, a threshold will be set. When the frequency of occurrence of a word or phrase less than the threshold, it will be replaced, and when the frequency of occurrence of a word or phrase is greater than the threshold, it will not be replaced.

Table 3.3: The example of the dictionary

weather	*sports*	*vehicles*	*crime*	*disease*	*career*
rain	tennis	school bus	gamble	mentally ill	police
gust	baseball	truck	prostitution	colon cancer	guard
downpour	running	jet	burglary	heart disease	actor
gush	basketball	van	murder	polyps	driver
deluge	football	car	larceny	rectal cancer	author
storm	swimming	bus	abduct	hypertension	doctor
wind	golf	subway	drug	tumors	director
snow	diving	train	kidnap	road rash	journalist
cloud	hockey	helicopter	rape	cystic fibrosis	deliveryman
tornado	boxing	bike	arson	diabetes	waitress

Algorithm 2 Pre-processing of Dictionary-based Generalization

Require: E' = {Weathers, Sports, Crimes, and Vehicles} , document d

1: Apply dictionary matching and the named entity recognizer to d and store the set of recognized (phrase,entity type) pairs to R'
2: **for** each (phrase, entity type) pairs $(w, e) \in R'$ **do**
3: Replace w in the documents with e
4: **end for**

- **AutoNER-based Generalization**:

  To replace the phrases and words more effectively, we have used AutoNER as the tool for automatic recognition. And expect that the AutoNER can improve the performance of the semantic content generalization on the pointer generator network.

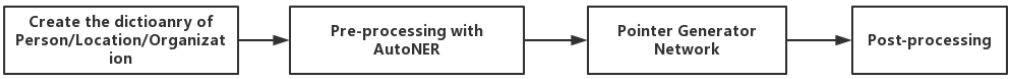

Figure 3.4: The flow of the AutoNER.

As can be seen from Figure 3.4, we will create the dictionary of "person", "location" and "organization" firstly, and then add them into the multiple categories dictionary for pre-processing. After pre-processing, we will input it into the pointer generator network to summarize and obtain the final output after post-processing.

About the dictionary of person, location, and organization, we used the Stanford NER to do the named entity recognition and extracted the phrases and entities from the result to compose the dictionary.

The input and output of the AutoNER is shown in Figure 3.5. We input the CNN/DailyMail datasets (the set of documents that are the target of summarization) and the wiki concept into the AutoPhrase [11] whose output will be used as the input of the AutoNER. The wiki concept is offered by Shang et al., and the purpose of the AutoPhrase is trying to extract high-quality phrases from datasets to make the dictionary richer. Then, we can obtain a text file that involved the full of high-quality phrases in the datasets and is called the full dictionary.

After that, we need to input the datasets, the created dictionary, and full dictionary into the AutoNER, and then we can obtain the labeled datasets.

3.2.2 Post-processing

Finally, we perform post-processing to obtain the final output summary. The post-processing used in this paper is the same as that proposed by Kouris et al. [9]. As they described, post-processing is a problem of the best binary matching. The matching is based on the similarity of the context around the generalized concept of summarization and the candidate concept of the text. If the generated concept is included in the taxonomy path, the generalized concept will be replaced with the word of the original text.

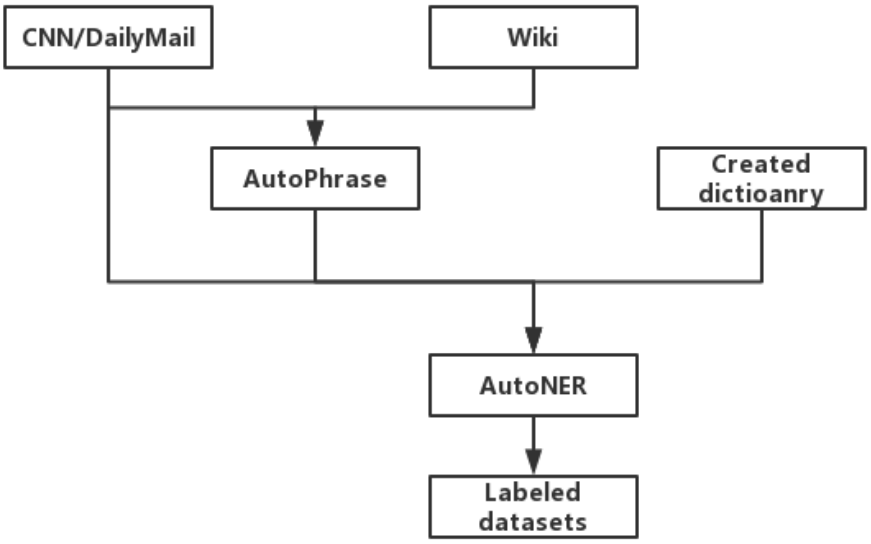

Figure 3.5: The input and output of the AutoNER.

Chapter 4

Experiments

4.1 Datasets

We used the multi-sentence summarization datasets CNN/DailyMail as the datasets, which is consist of the online news articles. The CNN datasets consist of more than 92,000 articles and corresponding summary, while the DailyMail datasets consist of more than 219,000 articles and corresponding summary. The training set is 287,226 training pairs, the validation set is 13,368 training pairs, and the test set is 11,490 training pairs.

We used the datasets which are the non-anonymized version of the data and do not pre-train the word embedding.

For the datasets, we apply the following process. All the words in the datasets are lowercase, and the numbers are replaced with #. We limited the maximum length of an article to 512. Due to the characteristics of the datasets, the length of the summary is uncertain, and the length of some summary is too long, which causes the excessive memory usage of neural networks, so the maximum length of the summarization is limited to 128. We use a vocabulary of 50, 000 words for both source and target. And truncate the article of the training set to 400 tokens, limiting the length of the summary to 100 tokens. At the same time, truncate the article of the test set to 400 tokens and limit the length of the summary to 120 tokens.

4.2 Settings

For the experiment, we used a pointer generator network with 256-dimensional hidden states and 128-dimensional word embedding. We used the coverage mechanism for anti-repetition. The other settings are the same as those of See et al. [1].

We didn't filter the phrases to be generalized based on its frequency, i.e., we set the threshold θ_f to be ∞, and it can also be understood as we have not set the threshold.

We apply the Stanford NER for named entity recognizer in the Algorithm 1 and Algorithm 2.

The baseline is the pointer generator network [1] with the coverage mechanism. It takes about a week to train the pointer generator network. When we apply the process for dictionary and NEG semantic content generalization, it takes about an extra three to four days compared with just training the pointer generator network. The server spec for the experiment is 56 vCPUs, 64GB RAM, 2 GPUs (GeForce GTX 1080 Ti).

4.3 Results

Although manual evaluation is the easiest way, to evaluate the automatic text summarization more efficiently, one or several metrics can be selected. Based on these metrics, we can compare the generated summary with a reference summary for automatic evaluation. At present, the most commonly used and most recognized metric is ROUGE. ROUGE is a set of metrics proposed by Lin [24]. We evaluated our model with ROUGE metrics, and report the F1 score of ROUGE-1, ROUGE-2, and ROUGE-L.

ROUGE-1 and ROUGE-2 are the kinds of the ROUGE-N. The formula of the ROUGE-N is like that:

$$ROUGE\text{-}N = \frac{\sum_{S \in ReferenceSummaries} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in ReferenceSummaries} \sum_{gram_n \in S} Count(gram_n)} \tag{4.1}$$

For example, we have reference summaries A and a generated summaries B below:
A: this dog is cute. B: this dog is so cute.
And we can get a table.

Table 4.1: The example of the ROUGE-N

	1-gram	*Reference 1-gram*	*2-gram*	*Reference 2-gram*
1	this	this	this dog	this dog
2	dog	dog	dog is	dog is
3	is	is	is so	is cute
4	so	cute	so cute	
5	cute			

And then, use the formula above, we can know the ROUGE score of the generated summaries.

$$ROUGE\text{-}1(A, B) = \frac{thisdogiscute}{thisdogiscute} = \frac{4}{4} = 1 \tag{4.2}$$

$$ROUGE\text{-}2(A, B) = \frac{thisdog, dogis}{thisdog, dogis, iscute} = \frac{2}{3} \tag{4.3}$$

ROUGE-L means that the longest common sub-sequence is abbreviated as LCS, and the formula is the following:

$$R_{lcs} = \frac{LCS(X, Y)}{m} \tag{4.4}$$

$$P_{lcs} = \frac{LCS(X, Y)}{n} \tag{4.5}$$

$$L_{lcs} = \frac{(1 + \beta^2) R_{lcs} P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}} \tag{4.6}$$

X is the generated summary, Y is the reference summary, m is the length of the generated summary, n is the length of the reference summary and LCS(X,Y) is the length of the longest common sub-sequence.

- **Named Entities-driven Generalization and Dictionary**:

  The result based on named entities-driven generalization and dictionary is shown in Table 4.2.

  The performance of the method that combines NEG with the pointer generator network is the highest excluding ROUGE-L. The scores of the ROUGE-1 can be improved by 0.0037 in the best settings. However, only when we compare the methods with the metrics of ROUGE-L, the semantic content generalization seems not to work well. The result of dictionary-based generalization is lower than the NEG, but the result of NEG with 3 categories is nearly the same as the result of NEG. And the result of NEG with 6 categories is higher than the result on the pointer generator. Therefore, from the result, we can see that the semantic content generalization can improve the pointer generator network.

  In the current, the dictionary only has a limited number of phrases and words, but the number of NEG is far larger than the dictionary. Our proposed method still can improve the performance of the pointer generator. It is furthermore proved that the effectiveness of semantic content generalization.

  To improve the ability of the semantic content generalization improvement, not only extend the number of the dictionary, but the processing methods of the dictionary also needs to be improved.

Table 4.2: The result on CNN/DailyMail with NEG and dictionary

	ROUGE-1	*ROUGE-2*	*ROUGE-L*
Pointer Generator	0.3824	0.1585	0.3205
Pointer Generator+NEG	0.3861	0.1592	0.2587
Pointer Generator+NEG+3 dictionaries	0.3860	0.1578	0.2575
Pointer Generator+NEG+6 dictionaries	0.3829	0.1558	0.2565

To verify the effect of post-processing on the results, we show the scores without post-processing in Table 4.3. The result without post-processing compared generated summarization with the replaced reference summarization. The result without the post-processing is not fair, and the purpose of it is trying to find that whether the post-processing works well or not, and whether restore the right phrase. Therefore, it is reference data for the result analysis. From Table 4.3, we can know that the score of the result without post-processing is overall higher than the score of the result with post-processing. From the results, the post-process causes a lot of errors by matching wrong words with the highest similarity. So, we need to improve the post-processing, since it is considered to be one of the major factors that deteriorate the performance of semantic content generalization.

The examples of experimental results are shown in Table 4.4. The first line in the table is the text of the reference summarization in the datasets. The second line is the result of the summarization of the pointer generator network. The third line is the result of the pointer generator network with NEG processing. It can be seen from the example text that if only NEG is used, the summarization will tend to be shorted. The fourth line is the result of the pointer generator network after NEG and 3 categories of

Table 4.3: The result on CNN/DailyMail with NEG and dictionary without post-processing

	ROUGE-1	*ROUGE-2*	*ROUGE-L*
Pointer Generator	0.3824	0.1585	0.3205
Pointer Generator+NEG	0.4146	0.1725	0.3517
Pointer Generator+NEG+3 dictionaries	0.4189	0.1752	0.3549
Pointer Generator+NEG+6 dictionaries	0.4171	0.1745	0.3544

Table 4.4: The example of the result on CNN/DailyMail with NEG and dictionary

Model	*Summaries*
datasets	foreign ministers of the united states france and germany cancelled travel plans for the next few days. aim to push for accord that would lay foundations for final settlement with tehran by the end of june. iran denies charges from the west that it wants to build nuclear weapon. it wants removal of international sanctions that are hurting its economy.
pointer generator	foreign ministers of the united states france and germany cancelled their travel plans for the next few days so they can push for the accord that would lay the foundations for a final settlement with tehran by the end of june.
pointer generator+ NEG	world powers intensified nuclear talks with iran yesterday. two days before a deadline for reaching a framework deal the foreign ministers of france states france and germany cancelled their travel plans for the next few days. so they can push for the accord that would lay the foundations for a final settlement with tehran.
pointer generator+ NEG+ 3 dictionaries	world powers intensified nuclear talks with iran yesterday. two days before a deadline for reaching a framework deal the foreign ministers of the united lausanne united and germany cancelled their travel plans for the next few days. so they can push for the accord that would lay the foundations for a final settlement with tehran by the end of june. us secretary of city also confirmed the french and german foreign ministers laurent fabius and walter walter steinmeier had cancelled a planned joint trip to kazakhstan.
pointer generator+ NEG+ 6 dictionaries	world powers intensified nuclear talks with iran yesterday. two days before a deadline for reaching a framework deal the foreign ministers of the united lausanne united and germany cancelled their travel plans for the next few days. so they can push for the accord that would lay the foundations for a final settlement with tehran by the end of june.

dictionary pre-processing. The last line is the result of the pointer generator network after NEG and 6 categories of dictionary pre-processing.

From the example text, no matter which way, it all can summarize the article accurately. For example text, "nuclear" is one of the keywords. Although for the pre-processing, the words "nuclear" does not belong to the items in the NEG and the dictionary, it is still retained due to the understanding of the overall semantics content. However, the pointer generator network neglect the keyword "nuclear", but the pointer generator with pre-processing can notice the keyword "nuclear". It also can prove the importance of semantic content generalization.

The summarization on the pointer generator network with NEG and 3 categories dictionary is the longest, and the summarization on pointer generator is the shortest. It is considered that the pointer generator network with NEG and 3 categories dictionary tend to remain the organization name and the person name which result in its length of the summarization is the longest. To solve this problem, the threshold is necessary. With the threshold, the pre-processing will not replace all words and phrases and then can help relieve this situation.

- **AutoNER and Dictionary**:

  The result based on AutoNER and dictionary is shown in Table 4.5. The performance of the pointer generator with AutoNER is not as well as our expectations. All the result is lower than the baseline (pointer generator network). And among the 3 types of the dictionary (3 categories, 6 categories, and 9 categories), the best one is the 6 categories. Combine with the result of named entities-driven generalization and dictionary, it proves that it is not the more categories of the dictionary, the better summarization performance.

Table 4.5: The result on CNN/DailyMail with AutonNER

	ROUGE-1	*ROUGE-2*	*ROUGE-L*
Pointer Generator	0.3824	0.1585	0.3205
Pointer Generator+AutoNER with 3 dictionaries	0.3277	0.0955	0.2056
Pointer Generator+AutoNER with 6 dictionaries	0.3289	0.0985	0.2102
Pointer Generator+AutoNER with 9 dictionaries	0.3154	0.0900	0.1992

For the result without the post-processing, compared to NEG and dictionary's slightly higher than the pointer generator network, the AutoNER is quite higher than the pointer generator network. And the result with AutoNER without post-processing is shown in Table 4.6.

Table 4.6: The result on CNN/DailyMail with AutonNER without post-processing

	ROUGE-1	*ROUGE-2*	*ROUGE-L*
Pointer Generator	0.3824	0.1585	0.3205
Pointer Generator+AutoNER with 3 dictionaries	0.5158	0.2375	0.4545
Pointer Generator+AutoNER with 6 dictionaries	0.5121	0.2318	0.4458
Pointer Generator+AutoNER with 9 dictionaries	0.5059	0.2262	0.4400

The examples of experimental results with AutoNER are shown in Table 4.7. The first line in the table is the text of the reference summarization in the datasets. The second line is the result of the summarization of the pointer generator network. The third line is the result of the pointer generator network with AutoNER (3 categories dictionary) processing. The fourth line is the result of the pointer generator network with AutoNER (6 categories dictionary) pre-processing. The last line is the result of the pointer generator network with AutoNER (9 categories dictionary) pre-processing.

If a word is not in the dataset or the high-quality phrase text file which is the output of AutoPhrase, AutoNER might not recognize the words. When the AutoNER can not recognize the words it will be identified as the unknown word and labeled with [UNK]. It can be seen from the example text that the AutoNER tends to appear [UNK] more easily. And it replaced almost all words and phrases in the summary and the phrases that the AutoNER extracted are quite strange. It may be the combination of the verb and noun and which is recognized as the location, organization, or even person. Besides that, the post-processing restores the phrases incorrectly. For example, when a person's name has two words, the post-processing will restore two words but the same word. Although the summary is not natural, it is still possible to summarize part of the article to a certain extent. And it is also necessary to set a threshold for AutoNER, to prevent from replacing almost all words and phrases in the article.

Table 4.7: The example of the result on CNN/DailyMail with AutoNER

Model	*Summaries*
datasets	foreign ministers of the united states france and germany cancelled travel plans for the next few days. aim to push for accord that would lay foundations for final settlement with tehran by the end of june. iran denies charges from the west that it wants to build nuclear weapon. it wants removal of international sanctions that are hurting its economy.
pointer generator	foreign ministers of the united states france and germany cancelled their travel plans for the next few days so they can push for the accord that would lay the foundations for a final settlement with tehran by the end of june.
pointer generator+ AutoNER with 3 dictionaries	[UNK] sunday talks nuclear talks with iran yesterday two days before a deadline for framework a framework deal the foreign officials of the united lausanne lausanne and britain cancelled their travel plans for the next few days so they can push for the accord that would lay the foundations for a deadline tehran with settlement by the end of june us kerry of state secretary kerry 's spokeswoman said he would not boston to reach for a ceremony in ceremony the late edward kennedy kennedy who was his officials officials close to the talks in the swiss city of end.
pointer generator+ AutoNER with 6 dictionaries	key programme intensified intensified talks with iran yesterday two days before a deadline for framework a framework deal the foreign deal of the united lausanne lausanne and from cancelled their travel plans for the next few days so they can push for the accord that would lay the foundations for a tehran tehran with settlement by the end of june us kerry of state secretary kerry 's spokeswoman said he would not fly to boston for a ceremony in england.
pointer generator+ AutoNER with 9 dictionaries	[UNK] ones intensified nuclear intensified with iran yesterday two days before a deadline for framework a framework deal the foreign the of the united lausanne lausanne and counterparts cancelled their travel plans for the next few days so they can push for the accord that would lay the foundations for a final tehran.

Chapter 5

Discussion

We tested and verified the effect of semantic content generalization on the pointer generator network, and applied AutoNER to the pointer generator.

From the result, we can know that semantic content generalization can improve the pointer generator network though our dictionary is limited. Therefore, not only to improve our dictionary, we should try to use the full version NEG to do the pre-processing rather than the simplified one (currently, we have only used a part of NEG and its post-processing) to furthermore improve the performance of the semantic content generalization. The full version NEG [9] has the threshold and it not only uses the Stanford NER [17], but also conjunction Stanford NER with the WordNet [18]. It will use Stanford NER to recognize the datasets and extract a part of named entities firstly, and then extract the other named entities from WordNet.

The result of ROUGE-L is lower than the pointer generator network. Because the ROUGE-L is the longest common sub-sequence, the low score can be explained that the result of the summaries has few longest common sub-sequence that consistent with the ground truth. How to improve the consistency with the ground truth should be discussed in future work.

However, from the result, the effectiveness of AutoNER still can not be verified. And we should consider the other methods that combine AutoNER with the pointer generator network.

Through the analysis of experimental results, we found that the final output summary often has matching errors which are caused by the post-processing. At the same time, the pre-processing with named entities-driven generalization and dictionary tend to remain the organization name and person name though those contents are possibly not important parts which need not be summarized. To address these, we should set a threshold in pre-processing to replace some phrases or words instead of replacing all the categories contained in datasets. Because the result without post-processing is better than the result with it, improving the post-processing will be one of our future research topics.

The AutoNER [12] has not performed well in the experiment, and we considered that the most probable reason maybe is the limited dictionary and the influence of the wiki dictionary. Firstly, we obtain a high-quality phrase text file through AutoPhrase [11], which is the output of the AutoPhrase used in the CNN/DailyMail datasets and wiki dictionary. However, the wiki dictionary is maybe not suited for our datasets. Secondly, AutoNER is the method based on the dictionary. But our dictionary is really limited, which is only 796

items in the dictionary. And these factors will indeed affect the effectiveness of the AutoNER at a certain level. Therefore, if we consider improving the performance of the AutoNER, we need to improve our dictionary and consider the alternative of the wiki dictionary that is offered by Shang et al. [11] currently.

Chapter 6

Conclusion

We examined the effect of the semantic content generalization method on the pointer generator network and through two different pre-processing. Besides that, we proposed two methods for semantic content generalization. The first method is named entities-driven generalization and dictionaries. From the result of the named entities-driven generalization and dictionaries, we can know that the semantic content generalization can indeed improve the performance of the pointer generator. The scores of the ROUGE-1 can be improved by 0.0037 in the best settings. And the reason why ROUGE-2 and ROUGE-L lower than the pointer generator had been explained in chapter 4 and chapter 5. The second method is AutoNER-based generalization. From the result of AutoNER-based generalization, we can know that the effectiveness is not as well as the first method of pre-processing.

We analyzed the reason why AutoNER-based generalization can not work well in chapter 4 and chapter 5, and we can obtain the conclusion below:

1. The semantic content generalization can improve the pointer generator network.
2. There are three reasons considered that the AutoNER can not effectively improve the pointer generator network:

   (1)The wiki dictionary is not suited for the CNN/DailyMail,

   (2)Too many phrases are replaced,

   (3)Recognition precision of the named entities is too low.
3. The matching precision of the post-processing is quite low.

In conclusion, through the experiment, we successfully verified our idea. Even though we used the limited dictionary, we still can improve the performance of the pointer generator network.

In future research, we will do more study on it to improve the performance.

1. To use full version NEG instead of the simplified NEG, which includes the use of the threshold.
2. To consider the alternative method for post-processing.
3. To find out the method that improves the ROUGE-L scores, which means improve the consistency with the ground truth.

Acknowledgement

I thank my supervisor Wakabayashi sensei who helps me a lot in many aspects. As an international student, my Japanese is not very well, so I can not understand what the Wakabayashi sensei means in the meeting. But I feel thankful that my Wakabayashi sensei is a kind person, he will explain it to me until I know how to do it.

Because of the coronavirus, I have a long time not to speak in Japanese, so my Japanese become strange, and can not understand if the Japanese speak too fast. So, when the intermediate presentation, my Wakabayashi sensei as the moderator remind the audience when they raise the question can speak slowly so that I can understand.

I think it is fantastic to be able to study in Japan, and I'm very thankful that my supervisor is Wakabayashi sensei. Can study under the Wakabayashi sennsei I think that studying not a stressful thing, it can also be interesting. And attend the regular meeting every week with Wakabayashi sensei also can be a joyful time.

It is a wonderful two years of study with Wakabayashi sensei. I am also very grateful to Wakabayashi sensei for the tolerance and understanding.

References

[1] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 1073–1083, 2017.

[2] Guanqin Chen. Chinese short text summary generation model integrating multi-level semantic information. In *2018 International Conference on Network, Communication, Computer Engineering (NCCE 2018)*. Atlantis Press, 2018.

[3] Dragomir R Radev, Eduard Hovy, and Kathleen McKeown. Introduction to the special issue on summarization. *Computational linguistics*, 28(4):399–408, 2002.

[4] Dipanjan Das and Andre FT Martins. A survey on automatic text summarization. language technologies institute, 2007.

[5] SongShengli, HuangHaitao, and RuanTongxiao. Abstractive text summarization using lstm-cnn based deep learning. *Multimedia Tools and Applications*, 78:857–875, 2019.

[6] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, pages 3104–3112. 2014.

[7] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing 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.

[8] Ramesh Nallapati, Bing Xiang, and Bowen Zhou. Sequence-to-sequence rnns for text summarization. 2016.

[9] Panagiotis Kouris, Georgios Alexandridis, and Andreas Stafylopatis. Abstractive text summarization based on deep learning and semantic content generalization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5082–5092, 2019.

[10] Jingwei Cheng, Fu Zhang, and Xuyang Guo. A syntax-augmented and headline-aware neural text summarization method. *IEEE Access*, 8:218360–218371, 2020.

[11] Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, and Jiawei Han. Automated phrase mining from massive text corpora. *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[12] Jingbo Shang, Liyuan Liu, Xiang Ren, Xiaotao Gu, Teng Ren, and Jiawei Han. Learning named entity tagger using domain-specific dictionary. In *EMNLP*, 2018.

[13] Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10, 2015.

[14] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, 2016.

[15] Kristina Toutanvoa and Christopher D. Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 63–70, 2000.

[16] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 252–259, 2003.

[17] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 363–370, 2005.

[18] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.

[19] Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. Language, Speech, and Communication. MIT Press, Cambridge, MA, 1998.

[20] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[21] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in neural information processing systems*, pages 2692–2700, 2015.

[22] Kristina Toutanova, Dan Klein, Christopher Manning, William Morgan, Anna Rafferty, Michel Galley, and John Bauer. Stanford log-linear part-of-speech tagger. *The Stanford Natural Language Processing Group, Stanford University Std.*, 2000.

[23] Fadi Hassan, Josep Domingo-Ferrer, and Jordi Soria-Comas. Anonymization of unstructured data via named-entity recognition. In *Modeling Decisions for Artificial Intelligence*, pages 296–305, 2018.

[24] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, 2004.