博士学位論文

A Study on Classification Problems in Natural Language Processing through Distributed Representation

分散表現による自然言語処理に関する分類問題の研究

東京工科大学大学院
バイオ・情報メディア研究科
コンピュータサイエンス専攻

令和 元 年 9 月

靳 展

Abstract

The essence of many research topics in the field of natural language processing is to solve specific classification problems. Existing methods tend to combine the distributed representation of text with rich syntactic and grammatical information to construct feature vectors for classification. The existing method can improve the accuracy of the classification result, but it will bring a variety of negative effects, such as reduced generalization, increased computing costs, and so on. The purpose of this study is to obtain a classification result similar to the existing method by using only the distributed representation of the text data in solving the classification problem of natural language processing. In the first part of the study, we found that the one-hot representation of a word is not satisfactory in the classification accuracy of the semantic relationship. We consider that this is because the one-hot representation of the word that it does not carry enough information about the semantic relationship. In the second part of the study, we proposed the *substring vectors* based on the distributed representation of words to classify semantic relationships. Without any syntactic and grammatical features and external semantic relational databases, we obtain classification accuracy higher than most similar methods. We can conclude that the distributed representation of words carries sufficient information about the semantic relationship, and this information can be used reasonably and efficiently in some way such as *substring vector* to solve the problem of semantic relationship classification. In the third part of the study, we used a distributed representation of sentences to train a deep neural network classifier in the actual CSCL project, and obtained classification accuracy similar to that of human coder. It shows that distributed representations perform better when combined with deep neural networks as classifiers than traditional classifiers. Based on the conclusions of the three parts of this study, we conclude that The distributed representation of text has better classification results than traditional syntactic and grammatical features when solving classification problems. Through the proposed *substring vector*, the potential information related to the semantic classification owned by the distributed representation can be utilized efficiently, and the classification result higher than most similar studies can be obtained, If we use a deep neural network classifier, we can more effectively exploit the advantages of distributed representation in solving classification problems in the field of natural language processing.

Contents

List of Figures

List of Tables

Chapter 1

General Introduction

1.1 Research Background

In recent years, with the rapid popularization of mobile internet, intelligent hardware and internet of things (loT), the world data shows an exponential growth trend. Meanwhile, there are more and more advanced data analysis technologies such as Machine Learning and Neural Network, and so do their research and applications. It has become a research hotspot to obtain valuable data from massive and complex data in Natural Language Processing (NLP). The semantic relations between two words is one of these valuable data. They have been widely used in many tasks of Natural Language Processing (NLP), such as Word Sense Disambiguation (WSD), Paraphrase, Document Summarization and Machine Translation. At the same time, a rich and structured database of semantic relations can play a very good role in assisting various Artificial Intelligence System (AIS). For example, when we chatted with a robot, we talked about the topic of "fruit". There should be a definition that the relationship between "fruit" and "apple" is "hyponym - hypernym" and the relationship between "apple" and "peel" is "whole - part". If there is no such a relational semantics database as a support for the robot, the conversation cannot proceed smoothly. Therefore, it is necessary to use an effective system to extract the semantic relations between two words from massive text data. So that the Artificial Intelligence System can accomplish tasks more correctly and efficiently.

Semantic relations is the relationship between two or more words based on word meaning. In some places, this relationship is also called lexical relations. Our research focusing on this relationship, the term "semantic relations" was used in this thesis to avoid ambiguity. Many existing researches have used different specific criteria to classify the semantic relations between two words. Most of the classifications of semantic relations have two main relationships, "hyponym - hypernym" and "whole - part". For example, the classification of semantic relations in the famous semantic relational database WordNet is shown

in Table 3.1. In the Associative Concept Dictionary (ACD) proposed by Okamoto et al. [1], semantic relations are divided into seven kinds of concepts, as shown in Table 1.2.

Table 1.1: Noun relations in WordNet

Relation	Definition	Example
Hypernym	From concepts to superordinates	breakfast - meal
Hyponym	From concepts to subtypes	meal - lunch
Has - Member	From groups to their members	faculty - professor
Member - Of	From members to their groups	copilot - crew
Has - Part	From wholes to parts	table - leg
Part - Of	From parts to wholes	course - meal
Antonym	Opposites	leader - follower

Table 1.2: Stimulus and associative concepts in Associative Concept Dictionary

Relation	Stimulus Concept	Associative Concept
Hypernym	chair	furniture, thing
Hyponym	chair	rocking chair, sofa
Part/Material	car	engine, tyre
Attribute	dictionary	difficult, helpful
Synonym	corridor	hallway, gallery
Action	newspaper	read, buy
Situation	book	bookshop, library

There are usually two ways to construct semantic relational databases, manual and automatic. Okamoto et al. [1] used the manual way of associative experiments to construct the dictionary of associative concepts. In their method, a stimulus word such as “car” was given to the participants in the experiment, then a semantic relation such as “whole - part”. If the answer is “engine” by association, the semantic relation between “car” and “engine” is recorded as “whole - part” and stored in the dictionary of associative concepts. This completely manual approach has the advantage of high accuracy, but the efficiency is low when constructing dictionary. Semantic relational databases are usually large, and most of them were constructed by automatic way. Semantic relations are automatically extracted from the massive text data through some predefined extraction rules. For example, Sumida et al. [2] proposed a method to extract the “hyponym - hypernym” relationship from hierarchical layouts in Wikipedia articles .

Semantic relation classification is a very important topic in the research of automatic extraction of semantic relations. Because the accuracy of classification will directly affect the classification results, and further affect the quality of semantic relational database. Among many existing researches on relational classification, the most representative and general one is supervised classification using labeled data. Many experimental results have shown that this is a very reliable classification method, and in most cases, good classification results have been obtained. In supervised classification, feature based classification is the most frequently used method. In order to achieve a high level of accuracy, it is necessary to have a set of heuristic features that can effectively represent the relationship between two words. The commonly used features are part-of-speech tagging (POST), syntactic patterns, prepositions and so on. Many research results have shown that it can bring better classification results by using more abundant and high-quality features. More and more research even used external resources when constructing features. For example, the required data were extracted from WordNet data, Wikipedia data and Google n-grams data to build features. But this also increased the complexity of features, resulting in increased processing time, which made it difficult to simplify the complexity of features while improving classification results.

1.2 Previous Studies

In view of the problems described in the previous chapter, our previous research was based on the associative concept dictionary proposed by Okamoto et al. The research results of Sumida et al. have shown that the “hyponym - hypernym” relations can be automatically extracted by using the hierarchical layouts in Wikipedia article and high classification accuracy can be obtained. Figure 1.1 showed the hierarchical layouts of “Black tea” in Wikipedia article. Unfortunately, the same method was not available to the automatic extraction of the “whole - part” relations. Because there were few words whose relationship with the title was “whole - part”.

Through research, we found that there were a large number of words whose relationship with the title is “whole - part” in the text of Wikipedia articles. The method we proposed here was as follows: First, the associative concept dictionary made by Okamoto and others was used as teaching data and a large number of “stimulus words” and “associative words” where extracted from the text of Wikipedia articles as candidate words. Then, the classifier was trained using the teacher data through the machine learning technology. Finally, the classifier was used to classify the candidate objects. Through this method achieved the

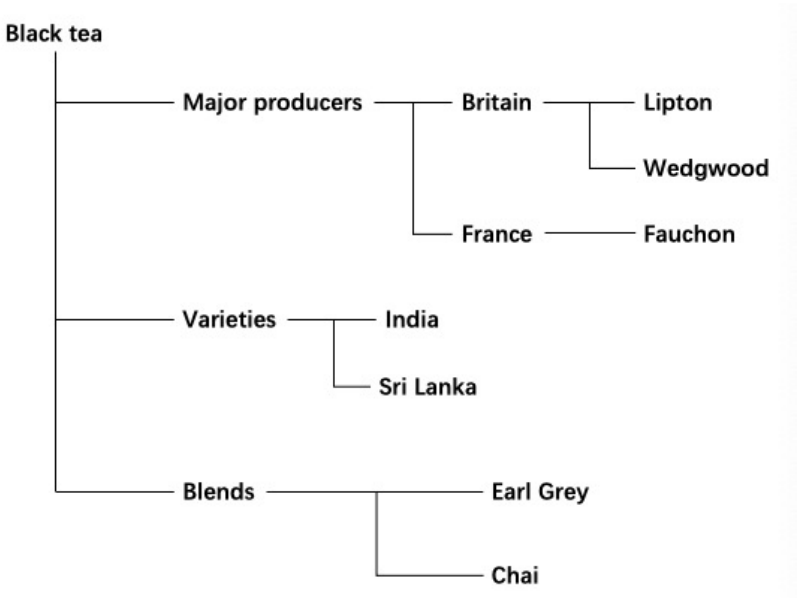

Figure 1.1: Hierarchical layouts of "Black tea" in Wikipedia article

purpose of automatically extracting "whole - part" relations from Wikipedia data to extend the dictionary of associative concepts.

1.3 Research Objective

After summarizing and analyzing the results of our previous studies, for the semantic relations extracted automatically, trained classifiers had a significant impact on the quality of relationships and the complexity of feature vectors had a significant impact on the efficiency of extraction. These two points referred to the last problem mentioned in the last chapter, which is, how to simplify the complexity of the features while improving the classification results. In order to solve this problem, there were three technologies that had attracted our attention: distributed representations for words, Neural Network Language Model (NNLM), and neural network that can be used as classifier.

Generally, there are two ways to represent a word by vector, one-hot representation and distributed representation. Representing a word as a vector with only one dimension is called the one-hot representation. For example, the word "apple" can be expressed as $[0, 0, 1, 0, 0, ...]$. The one-hot representation often faces the curse of dimensionality in practical applications. Take the probabilistic language model as an example. Assuming that the set of words is V, for the simplest trigram language model, the parameter space is $|V|^3$. Assuming that there are 100,000 words in the vocabulary of corpus, the parameter space is 10^{15}. This has far exceeded the computing power of ordinary computers. In the

meanwhile, because the value of most of the vector is 0, it also faces serious data sparsity issues. Moreover, because words are isolated, there is no way to reflect the potential semantic relations between words in the vector represented by one-hot. In order to overcome the shortcomings of one-hot representation in terms of semantic relations, Hinton et al. [3] first proposed the distributed representation of words. The distributed representation of words is also called the word embedding. To a certain extent, it can be used to describe the semantic distance between words. For example, the distributed representation of the word "apple" may be $[0.11, 0.77, 0.71, 0.10, 0.50, ...]$. The difference between one-hot representation and distributed representation is that the former uses a vector of one dimension, while the latter uses a dense real vector to represent a word. So the dimension of distributed representation is usually relatively low, usually around a few hundred. In practical applications, distributed representation can effectively alleviate the problem of data sparseness. Using distributed representation, we can efficiently calculate the semantic relations between words in low-dimensional vector space.

Unfortunately, the computation process of distributed representation of words is usually complicated. From the early latent semantic indexing to the recent neural network language model, researchers have developed various models to learn the distributed representation of words. Bengio et al. [4] did the relative researches of Neural Probability Language Models (NPLM) [5], which made distributed representation of words gain wide attention. Bengio et al. used a three-layer neural network to build a language model, as shown in Figure 1.2. But the early neural network language model like this was inefficient and difficult to be used in practice. On the basis of these early related studies, Mikolove et al. [6] proposed that a simpler network model was used to get the distributed representation of the word by using the context before and after the word. They simplified the non-linear hidden layer in the traditional neural network language models and developed two simpler neural network models, Continuous Bag-of-Words (CBOW) model and Skip-Gram model, as shown in Figure 1.3. The experimental results showed that the distributed representation of the words obtained through these two models was much better than that obtained from the traditional neural network language model and the training time was only about $1/10$ of the latter. Now, these two models have become the representative models of distributed representation of learning words. The distributed representation of the acquired words obtained by these two models after being trained through huge text datasets like Wikipedia must contain abundant semantic relations between words. This distributed representation of words can be used to construct feature vectors with high quality.

In our previous research, the traditional Support Vector Machine (SVM) classifier was used [7]. In recent years, many studies have shown that if the characteristics of data are

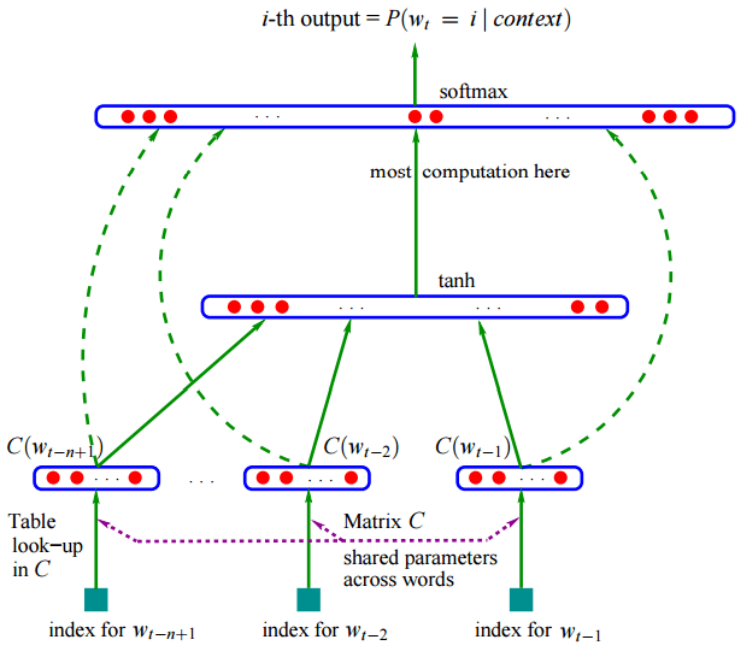

Figure 1.2: An overview of the network architecture of neural probabilistic language model

not obvious, when classifying potential relationships in data, the results obtained by the trained neural network is better than those of traditional classifiers. A single-layer fully-connected neural network for classification, input data is a feature vector constructed in advance according to a variety of features, and output data is the prediction of semantic relation classification. We can use the teacher data to train the neural network. During the training process, the weight and bias of the hidden node will be updated. Then the trained neural network can be used to predict the relationship between two unknown words.

According to the research of Zeng et al. [8], the classification results of semantic relations were not only related to the neural networks used, but they also had much to do with the feature set chosen when constructing feature vectors. Most of the existing researches combined several feature sets to construct feature vectors. Even some feature sets were extracted from external resources. Table 1.3 shows the commonly used external resources in the existing researches for semantic relations classification. Although the use of external resources can improve the accuracy of semantic classification, it also had obvious shortages, which made the algorithm in the research method more complex and certainly increased the number of dimension of feature vectors. It was closely related to the language types of the subjects. Once the language types of the subjects are changed, for example, from English to Japanese, the external resources used will be invalid.

For the problems in the previous research, The motivation of our research was to find a

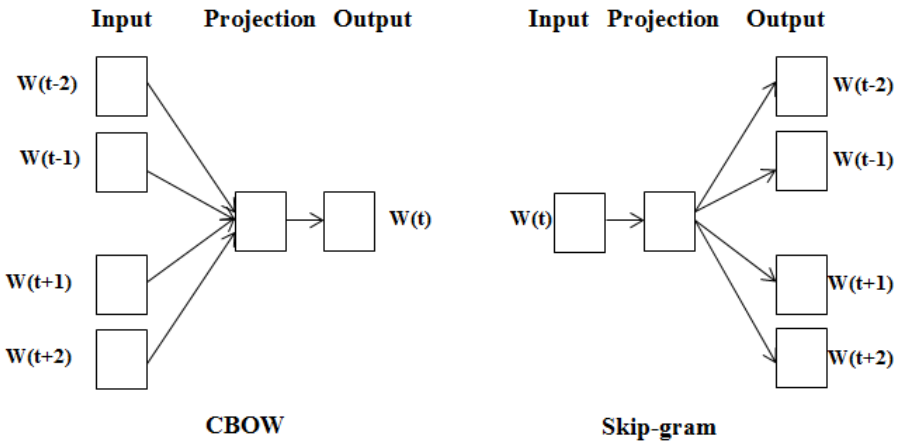

Figure 1.3: CBOW and Skip-gram neural architectures

Table 1.3: The most commonly used external resources for semantic relations classification

Resource Name	Description
WordNet	A lexical database of English words that are linked together by their semantic relationships.
Google n-gram	A dataset that Google have created for language modeling over time using text of books.
OpenCyc	The world's largest and most complete general knowledge base and commonsense reasoning engine.
PropBank	A corpus that is annotated with verbal propositions and their arguments.
NomBank	A project at New York University to annotate the argument structures for common nouns in the Penn Treebank II corpus.
Roget ' s Thesaurus	A widely used English-language thesaurus.

simple way to construct a low-dimensional feature set and use it to construct feature vectors, and finally obtain high-precision results of semantic relation classification. In other words, we needed to build a simple and intelligent distributed representation of text data that can contain as much potential grammatical and semantic information as possible. We were inspired by the research on distributed representation of words. According to the research of Fu et al. [9], the F-score of predicting the "hyponym - hypernym" relations was improved by using the distributed representation of words. Because the distributed representation of words contained a lot of semantically related information, we proposed to construct feature vectors based on the distributed representation of words. We introduced a new distributed representation of the partial word sequence between two words in a sen-

tence, which we called *substring vector*. We only used this new distributed representation feature set to classify semantic relations. Compared with similar researches, a higher accuracy (78.10%) was achieved in our research. Furthermore, unlike other similar studies, no external resource was used in our method. The advantages of this are: first, it did not rely on the language type of the target, which made it easy to apply to other similar tasks. Second, the number of dimensions of the feature vector was lower than that of the similar researches, which improved the classification efficiency. The novelty of our method is that we proposed a new simple and effective weighting method based on word frequency. According to the experiments, the F-score of the semantic relation classification results can be increased by 1% to 3% after using the weighting method.

1.4 Thesis Organization

In this thesis, the automatic extraction technologies of semantic relations was researched and a new *substring vector* was proposed to construct feature vectors for semantic classification in the aim of improving the accuracy and efficiency of semantic relation classification.

Chapter 2 introduced our previous research, which took the association concept dictionary as the research object. The study was aimed at automatically extracting association concepts from Wikipedia data. Chapter 3 introduced our research methods, which was the core of this thesis. We focused on the classification of semantic relations. The classification was done by using an efficient low-dimensional distributed representation of partial word sequences in text data, which was a lightweight way of processing. We proposed a method to construct classification features of semantic relations. This method consisted of only a low-dimensional vector of partial word sequences between two words. In addition, we also studied the relationship between the number of dimensions and accuracy when using a non-linear classifier. In the Chapter 4, through a specific research, we used a deep neural networks to solve the problem of classification of sentences in educational data. We generated a distributed representation of the sentence and then used it as a feature vector. These feature vectors are used to train a deep neural network, and finally the deep neural network is used to classify the labels of the educational data. We used the distributed representation of the sentence to solve this practical and specific problem. Chapter 5 summarized our research.

Chapter 2

Semantic Relation Extraction and Classification from Combination of Associative Concept Dictionary and Wikipedia Data

2.1 Introduction

When humans talk to each other about a topic, in order for the dialogue to proceed smoothly, it is necessary to use background knowledge about the topic. For example, when discussing the topic of "fruit", it will recall words like "apple", "orange", "health" and "price", and the semantic relations between these words and "fruit" is this background knowledge of the topic. How to let computers learn this background knowledge has become an important topic in the field of artificial intelligence and natural language processing. In particular, in order to understand the correct meaning of polysemy or homonym in the current sentence, we need to choose according to the background knowledge of the article. For example, when one sees the phrase "our manager is a devil", it is easy to understand that the manager is not a real devil, but wants to express that the manager is a strict person. But it is very difficult for a computer to understand the meaning of this sentence. If the computer is only interpreted according to the literal meaning, it will cause a lot of misunderstanding.

In order to solve the problems mentioned above, a database of associative concepts such as "devil - terror" is needed as support. Okamoto et al. [1] proposed a method to construct associative concept database. They obtained associative concepts through associative experiments. Firstly, the subjects were given a basic word, then seven kinds of semantic relations were given, such as "hypernym concept", "hyponym concept", "part/material concept". Finally, the words associated with each semantic relations were recorded. After a

large number of experiments, a large-scale dictionary of associative concepts has been constructed based on the recorded experimental data. At present, the size of the dictionary of associative concepts is: 1055 stimulus concepts, and about 250,000 associative concepts corresponding to stimulus concepts, which are increasing gradually. However, the dictionary content is all manually produced by researchers through association experiments. Compared with automatic extraction through the system, this original manual method has the advantages of higher accuracy, but the efficiency of manual production is very low. With the rapid development of various technologies, the concept of association of the same word will change with the development of the times. For example, for words like "phone" and "television", their associative concepts is vary greatly in different ages, and even many new associative concepts will emerge. Therefore, the low efficiency of manual mode becomes a more and more serious problem.

To solve above problems, our goal is to propose a method of automatically extracting associative concepts to expand the dictionary of associative concepts. Our research methods are as follows. Firstly, a large number of word pairs are extracted from a large number of text data on the network as a candidate for associative concepts. Then, using machine learning technology, the existing Associative Concept Dictionary is used as teacher data to train the automatic extraction system. Finally, the associative concepts which are correctly classified and do not exist in the existing Associative Concepts Dictionary are selected from the alternates, and added to the existing dictionary as new associative concepts, so as to expand the existing Associative Concepts Dictionary. In the existing research, there have been many studies on automatic extraction of "hypernym concept", "hyponym concept" and "synonym concept" which have been obtained satisfactory results. Due to the lack of appropriate teacher data, there are few studies on automatic extraction of "part/material concept". We believe that the existing dictionary of associative concepts contains a lot of appropriate teacher data related to "partial/material concept".

In this study, we select "part/material concept" as the research focus from seven associative concepts, automatically extract candidate associative concepts from a large number of text data such as Wikipedia articles, train SVM classifier with Associative Concept Dictionary as teacher data, and extract new "part/material concept" from candidate by classifier. The experimental results show that we have achieved very significant results.

2.2 Associative Concept Dictionary and Associative Experiments

In order to build an artificial intelligence system, we need to take full account of the background knowledge of words, which requires a large-scale and structured concept database. Existing representative conceptual databases include: WordNet in English [10], EDR Electronic Dictionary [11][12] and Associative Conceptual Dictionary in Japanese [1], etc.

Associative Concept Dictionary is a systematic database based on large-scale experimental data of association. It contains a large number of pairs of words composed of stimulus concept and multiple associative concepts associated with stimulus concept. The association distance between stimulus concept and associative concept is quantified. Associative experiment is a cognitive experiment designed to elucidate the structure of human knowledge. Specifically, it shows the subjects a stimulus concept and seven kinds of semantic relations, and records the associative words as associative concepts through the free association of the subjects. The concept of stimulus is a combination of the most basic nouns appearing in Japanese primary school textbooks and the nouns supplemented in experiments as a phrase of stimulus concept. Okamoto [13] defines the concept of association as seven kinds of semantic relations, i.e. “hypernym concept”, “hyponym concept”, “part/material concept”, “attribute concept”, “synonym concept”, “action concept” and ”situation concept”. Each semantic relations and example are shown in Table 1.2. For each stimulus concept, the associative contents of 50 subjects were recorded. At present, the concept of stimulation in associative concept dictionary has reached 1055, and associative concept is about 250 thousand. The dictionary is constantly being expanded through the associative experiment. At the same time, many studies on the construction of semantic network and the resolution of word ambiguity using Associative Concept Dictionary are under way [14].

Because the concept of association is constructed manually by researchers through associative experiment, there is a problem of low efficiency. At present, there have been many studies on automatic extraction of the two semantic relationships, and good results have been obtained, but there are few studies on automatic extraction of “part/material concept”. This study proposes a method of automatically extracting the concept of part/material from massive text data from the Internet.

2.3 Related Work

Among the seven semantic relationships in the Associative Concept Dictionary, there are many existing studies on the "hypernym concept", "hyponym concept" and "synonym concept", and the research results have been widely used. Hearst et al. [15] proposed a lexicon-syntactic patterns approach to extract hypernym and hyponym relations from newspaper articles. After that, with the development of the Internet, Shinzato et al. [16] proposed a fast and automatic method to extract natural language expressions with similar semantics from a large number of HTML articles on the Web. They proposed a method to classify words into several semantic classes. The experimental results were verified by four subjects, and the accuracy rate was 80%. In the latest research on the semantic relationship between words based on Wikipedia data, Sumida et al. [2] proposed a method to extract the "hypernym/hyponym concept" through hierarchical layouts of Wikipedia article. Firstly, they assumed that there are many "hypernym/hyponym concepts" in the hierarchical layouts of Wikipedia articles. Then, they use the data of entries and lists in Wikipedia articles to filter through machine learning. Finally, they got 1.35 million sets of word pairs of "hypernym/hyponym concepts". The experimental results showed that the correct rate of the results is more than 90%.

However, for the concept of part/material, the hypothesis of Sumida is not valid. There are very few "part/material concepts" in the hierarchical layouts of Wikipedia article. For example, as shown in Figure 1.1, the hierarchical layouts of Wikipedia article titled "coccinellidae", include words such as "coccinella septempunctata", "harmonia axyridis" and "menochilus sexmaculatus", which are semantically related to "coccinellidae" by hypernym or hyponym relation. But there are no words like "legs", "antennae" and "coccinellidae" that are semantically related to part/material relation. Through our research, we find that the title of Wikipedia article, part/material relation words mostly appear in the body of the article. For example, in the article titled "coccinellidae", there is a sentence "Coccinellids are often conspicuously coloured yellow, orange, or red with small black spots on their wing covers, with black legs, heads and antennae. ", in which, "covers", "legs", "heads", "antennae" and "coccinellidae" are semantic relations of part/material. And, these words are often juxtaposed in sentences.

If we know that there is a "partial/material concept" word pair like "coccinellidae - antennae" in the Associative Concept Dictionary, it is possible to find a "part/material concept" that does not exist in the dictionary, such as "coccinellidae - cover", "coccinellidae - leg" and "coccinellidae - head". So we propose a new hypothesis, if we know that there is a partial/material relation word pair w_s and w_r in the Associative Concept Dictionary (where

w_s is the stimulus concept, w_r is the associative concept), in the body of the Wikipedia article titled w_s, if a sentence contains w_r and there is w_{n1}, w_{n2}... which is a side-by-side relationship with w_r, then the possibility of the partial/material semantic relationship is very high for w_{n1}, w_{n2}... and w_s. Based on the above hypothesis, we propose a new method to automatically extract "partial/material concept" that do not exist in Associative Concept Dictionary using Wikipedia data. The validation experiments show that our proposed method is effective.

Since the Associative Concept Dictionary is a conceptual dictionary on Japanese, the Wikipedia article mentioned in this chapter is in Japanese.

2.4 Proposed Method

In this thesis, we use the text of the Wikipedia article as the data source and the Associative Concept Dictionary as the teaching data, and propose a new method that can automatically extract the "part/material concept". The details of the method are as follows.

First, we get the required word sets from the Wikipedia data based on the Associative Concept Dictionary. Since stimulating words and associative words in the Associative Concept Dictionary are one-to-many forms, we can get such a set $\{w_{s1} \mid w_{r1}, w_{r2}, ..., w_{rn}\}$ according to the dictionary, where w_{s1} is a stimulating word , w_{r1} is the associative word. We get the relevant text data titled w_{s1} from the Wikipedia data. Then we divide the text data into sentences, and separate the words in each sentence by morphological analysis to obtain words set W.

Second, we classify word collections according to the Associative Concept Dictionary. If a word w_x in the words set W exists in the set $\{w_{r1}, w_{r2}, ..., w_{rn}\}$, we extract the word, and after traversing the words set W, all the extracted words can form a words set X. According to our hypothesis, if a certain w_y in W and each element in the words set X are not in a side-by-side relationship, the probability that w_y is an associative word is very low. We extract words like w_y to form a words set Y. We extract the words in the words set W that do not belong to the words set X or the words set Y, and form a words set Z as a candidate for the new associative word.

Third, we create feature vectors for each of the words in the words set X, Y, and Z. We extract the syntax information of the word as a feature to construct a feature set. The feature set of the words set X is a positive example. The feature set of the words set Y is a negative example. The positive and negative examples are used as training data to train the classifier, and then the trained classifier can used to classify the word set Z to obtain new associative words.

The generation of training data and the extraction of features will be explained in detail in the following sections.

2.4.1 Pattern of Training Data

Table 2.1: The form of the training data

Data A	{ stimulus | associative }
Data B	{ stimulus | sentence }
Data C	{ stimulus | associative | sentence }
Data D	{ stimulus | associative | candidate associative | syntax information }
Data E	{ stimulus | associative | candidate associative | syntax information | label }

The form of the training data used to train the classifier is shown in Table 2.1. The training data is prepared by the following five steps show in Figure 2.1.

Step1. Extract the "part/material concept" data from the Associative Concept Dictionary. There are seven kinds of semantic relations in the Associative Concept Dictionary. We only extract the relevant data of "part/material concept". The stimulus words and the associative words in the dictionary appear in pairs, so a collection of "Data A" as shown in Table 2.1 is obtained.

Step2. Extract text data from Wikipedia articles in units of sentences. Using the stimulus word in the Associative Concept Dictionary as the title of the article. In order to facilitate the processing of data, we use the Wikipedia API to get the required text data and save it in the local database. After that, we divide the text data into sentences, and obtain a set of "Data B" as shown in Table 2.1.

Step3. Integrate the collection of "Data A" and "Data B". We integrate the data of the same word in the set of "Data A" and "Data B" to obtain the set of "Data C" in Table 2.1. After that, we traverse the words in the sentence of "Data C". If the associative word exists in the sentence, then the data is retained as valid data, and other data is discarded as invalid data. The purpose of this is to reduce the number of negative examples as much as possible.

Step4. Add syntactic information to the collection of "Data C". We use the morphological analyzer to parse the sentences in "Data C", and divide the sentences in "Data C" into words, and finally add the syntactic information obtained after parsing to the end of each data. Thus, the set of "Data D" in Table 2.1 is obtained. After this step, the smallest unit in the data is changed from one sentence to one word. and one "Data C" is split into several "Data D".

Step5. Add positive and negative labels to the “Data D”. In order to train classifiers such as SVM, “Data D” needs to be divided into positive and negative examples. For a “Data D”, if its alternate word is the same as the associative word, then it is considered a positive case (labeled as +1.0). If its alternate word and the associative word are belong to different category and are not juxtaposed with the associative word, then it is considered to be a negative example (labeled as −1.0). In this way we obtain the “Data E” set in Table 2.1. Since we have no way to directly generate negative examples based on the Associative Concept Dictionary, we can only indirectly complete the construction of the negative examples.

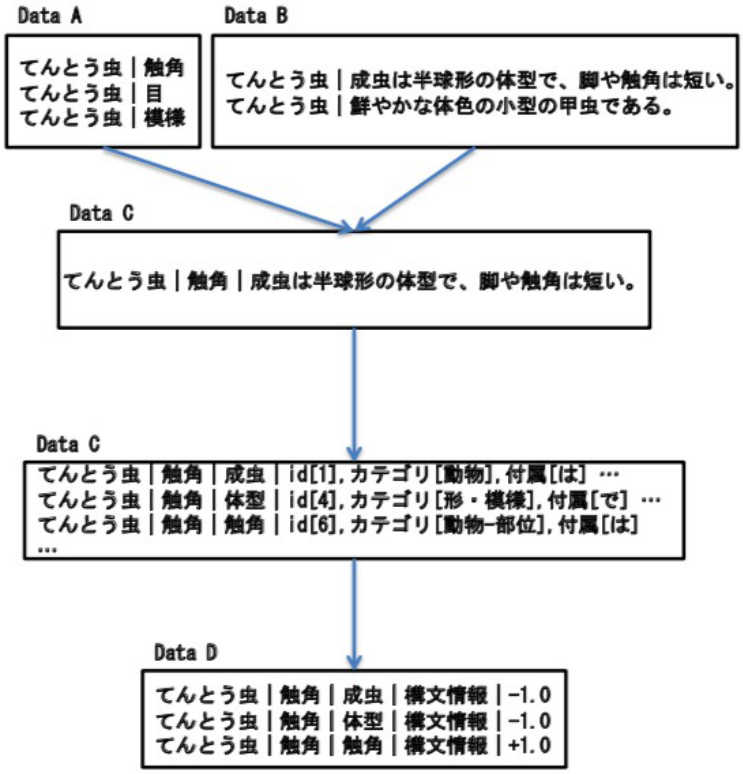

Figure 2.1: The preparation process of training data

2.4.2 Extraction of Features

In general, how to construct the feature vector of the training classifier will have a great impact on the precision of classification. So we should choose the most effective feature as far as possible. In this study, for a word w, we choose two features as follows: First, we take the category list of the word w as a feature. Second, we take the dependency list

of the word w (including the Japanese auxiliary word) as a feature. The feature vector is generated by the following three steps.

Step1. The syntactic information of the words in the positive example of the training data is extracted to generate a category sequence K and a dependency sequence F.

Category List : $K = [k_1, k_2, k_3, \cdots, k_n]$

Dependency List: $F = [f_1, f_2, f_3, \cdots, f_m]$

Step2. The category sequence K and the dependency sequence F are combined together as a feature set S.

Feature Set: $S = [k_1, k_2, k_3, \cdots, k_n, f_1, f_2, f_3, \cdots, f_m]$

Step3. The syntactic information of each word is compared with the feature set to create a feature vector that can be used to train the classifier. The value of the feature vector is binary, $True(1.0)$ and $False(0.0)$. For example, if the word w belongs to the category k_1 and there is a dependent word f_2 in the sentence x, then the feature vector of the word w is $[1.0, 0.0, 0.0, ..., 0.0, 0.0, 1.0, 0.0, ..., 0.0]$. We append the feature vector to the positive and negative labels of the previous section "Data E" to form the following data form:

{ stimulus | associative | candidate | syntax information | label | feature vectors}

The feature vector here is also the input vector of the classifier.

2.5 Experiment Results

2.5.1 Tools and Dataset

In order to evaluate the effectiveness of our method, we selected 200 stimulus words from the Associative Concept Dictionary to conduct a validation experiment. In this experiment, we used the Japanese Dependency and Case Structure Analyzer KNP to parse sentences from the Wikipedia articles. We applied three kinds of classifiers to our data: (1) a non-linear SVM with the Gaussian radial basis function kernel (SVM-RBF), (2) a linear SVM (SVM-LN), and (3) a simple feed-forward neural network with a two-way softmax classifier (NN+softmax).

We now describe the experimental steps in details. Firstly, we selected 200 stimulus words from the existing Associative Concept Dictionary as the object of this experiment. Since the stimulus words and the associated words in the dictionary are one-to-many relationships, we obtained 42,452 "Data A". The semantic relationship between the stimulus word and the associated word was part/material relations. Secondly, 200 stimulus words were used as the title to obtain text data from Wikipedia articles, and text data was segmented in units of sentences. After eliminating duplicate and invalid data, 49,397 "Data B" valid data was left. Thirdly, we combined "Data A" and "Data B", and filtered out the

sentences containing the associated words in “Data B”, 16,611 “Data C” were generated. Fourthly, we used the tool KNP to parse the sentences in “Data C” and divided into “Data D” according to the obtained syntactic information. Since KNP also has the morphological analysis function, we can easily extract each word in a sentence. The “Data C” was segmented in units of words, 19,910 “Data D” were generated. Finally, according to the criteria of labels, “Data D” was divided into 489 positive examples and 19,421 negative examples. The number of intermediate data is shown in Table 3.5.

Table 2.2: The number of intermediate data

Data A	42,452
Data B	49,397
Data C	16,611
Data D	19,910
Positive Examples of Data E	489
Negative Examples of Data E	19,421

Based on the method described in Section 2.4.2, we obtained a feature vector with dimension 2,885, in which the number of category features was 146 and the number of dependent words was 2,739.

According to the pre-processing results of the experimental data, the number of negative examples is approximately 40 times the number of positive examples. This is a typical imbalanced data problem in machine learning. Training models using such imbalanced data will be skewed towards the minority class. We used a traditional and effective method to solve this problem, randomly extracting the same amount of positive data from the negative data. We combined positive data and negative data as our experimental data, 80% of the data as training data and 20% as verification data. The composition of experimental data set is shown in Table 2.3

Table 2.3: The composition of experimental data

Number of Experimental Data	978
Number of Training Data	782
Number of Verification Data	196

In this experiment, we applied three kinds of classifiers for classification. For the parameter setting of SVM-RBF, the kernel type was set to radial basis function, penalty parameter C was set to 1.0, gamma was set to 0.1, and degree was set to 3. For the parameter

setting of SVM-LN, the kernel type was set to linear, and other parameter setting was similar to SVM-RBF. For the neural network, it was a four-layer neural network. It had two hidden layers, and the number of neurons in the two hidden layers is 1,400 and 500 respectively. We set the learning rate to 0.001 and the maximum number of training epochs to 100.

2.5.2 Results and Discussion

Table 2.4: The classification performance of three classifiers

Classifier	Accuracy	Precision	Recall	F-score
SVM-LN	0.678	0.698	0.679	0.670
SVM-RBF	0.688	0.716	0.689	0.679
NN+softmax	0.740	0.743	0.740	0.739

Table 2.5: The confusion matrix for validation data

		Actual(Associative Concept Dictionary)	
Predicted (NN + softmax)		Positive	Negative
	Positive	67 (TP)	21 (FP)
	Negative	20 (FN)	78 (TN)

The accuracy, precision, recall and F-score of classification for the three classifiers are shown in Table 2.4. We used the training data to train the neural network, and then used it to predict the verification data. The confusion matrix of the prediction results is shown in Table 2.5. According to Table 2.4, we knew that the classification result of a simple neural network is better than the traditional SVM classifier, and according to experience, with the optimization of the neural network, the classification result can be improved.

Based on the analysis of the prediction error data, we found that it has a lot of bad influence on the classification results because the number of negative samples in the original data is much larger than the number of positive samples. For example, for the word "forest", words such as "water" and "walking" appear in sentences more often than "part/material concept" words, such as "trees". Therefore, in this case, even if the probability of such an error occurring in the "negative example being misjudged as a positive example" is small, since the number of negative examples is large, and comparing with the number of "correctly determined in the positive example", the ratio of misjudgment results is also

very high. Even if we use random sampling to artificially reduce the number of negative samples, it does not improve the situation, and unfortunately, it also causes loss of features in the negative case. If we use a more precise rule to determine the negative sample, we can get better classification results. Through experiments, we found that since not every word has a category feature, it has less contribution to the classification features. At the same time, we also found that in the training model, the feature of most dependency words have weights, which also indicates that the dependency words have a positive influence on the classification results. From this we believe that the dependency word is a very effective feature in the semantic relationship of "part/material concept".

2.6 Conclusion

In this study, we proposed a method for automatically extracting "part/material concept" using the Associative Concept Dictionary and Wikipedia data. For "part/material concept", we combined the Associative Concept Dictionary and Wikipedia data to build training data, train the classifier, and then use the trained classifier to validate our method against 196 pieces of validation data. The accuracy of classification results is 74%.

Although there are already many studies on "hypernym concept" and "hyponym concept" which have obtained a high accuracy, the existing research is less involved in the "part/material concept". In this study, we took "part/material concept" as the research object and obtained very meaningful results.

We list the following three aspects as topics for improving accuracy in the future. Firstly, the data used in this study is only a small part of the Associative Concept Dictionary, which only including the data related to "part/material concept". If we can use data from other association concepts, it will improve the precision of the results to some extent although the number of sample points will increase significantly. Next, besides category information and dependency information, there is much other more detailed information related to words, such as the position of words, etc. If this information can be used as features, we can enrich the types of features and obtain better precision. Finally, we believe that if using deep neural networks instead of the simple neural networks used in our experiments, the better classification results will be obtained.

Chapter 3

Semantic Relation Classification through Distributed Representations of Partial Word Sequences

3.1 Introduction

The popularity of the internet and computers, data now become massive and public in our society. In recent years, beyond the notion of big data, representation of data that is more efficient for intelligent systems, is received considerable attention. Semantic relations between two words extracted automatically from text data are in such type of data. They are widely available for many tasks of Natural Language Processing (NLP) applications, such as Word Sense Disambiguation (WSD) [17], Paraphrasing [18], Document Summarization [19] and Machine Translation [20]. For instance, if we chat with robots and they cannot extract semantic relations from our words, the conversation does not smoothly proceed. Thus, an efficient semantic relation classification mechanism is required to obtain the background knowledge of robots [21][22], so that data can be applied more smartly on existing intelligent systems.

In the past few years, relation classification has attracted considerable research interest. Many approaches have studied relation classification, the most representative and general one is that of supervised classification from labeled data, which has been shown to be reliable and yield good classification results in most cases [23][24][25][26]. In supervised classification, feature-based approaches are used most commonly. In these approaches, to achieve high-level accuracy, a set of heuristic features that can effectively represent relations between two nominals must be determined. Since using richer and higher-quality features leads to a better performance, in existing approaches, frequently various features, such as part-of-speech tagging (POST), syntactic patterns, and prepositions [27][28], are used

and external resources, such as WordNet data, Wikipedia data, and Google n-grams [29], are imported. This tendency also increases the complexity of the feature set which resulting in an increased processing time. Thus, it is difficult to simplify the complexity of the features and improve classification results simultaneously.

To solve these problems, there are two techniques that have received considerable attention: distributed representations for Words and Neural Network Language Models (NNLMs). In general, an appropriate set of vectors in $\mathbb{R}^N$, where words are mapped and N is sufficiently small, e.g., less than several hundreds, and the elements of which are usually not zero, is called a *distributed representation* of words [30]. Zeng et al. (2014) [8] showed that using the Convolutional Deep Neural Network (CDNN) model with lexical and sentence-level features yields better results than other existing approaches. In the CDNN model, as well as other NNLMs, words must be converted into vectors through some distributed representation before the model is applied.

An appropriate distributed representation is the key point for achieving highly accurate classification results. Most of the existing methods are combined with several feature sets to construct the appropriate distributed representation. Even some feature sets are extracted from external resources. Table 3.1 shows number of external resources and feature sets in the existing research. Although the use external sources can improve classification accuracy. It has obvious drawbacks; it makes the algorithm of approach more complex, and increase the number of dimension of the feature vectors. If the object language has been changed, the external resources used in the approach will be invalidated. Our motivation is to find a simple way to build a simple and small feature set, and to use only that feature set to obtain high classification accuracy. In other words, we need a simple and smart representation of data that contains a considerable amount of potential syntactic and semantic information.

We obtain the inspiration from the related research on word vectors. Distributed representations of words called *word vectors* proposed by Mikolov have shown to preserve linguistic regularities, such as the semantic relations between two words [6][31]. For example, it is known that $v(\text{king}) - v(\text{queen}) \approx v(\text{man}) - v(\text{woman})$, where $v(w)$ is the vectors of the word w. Fu et al. (2014) showed that the prediction performance for the hypernym-hyponym relation is improved in terms of F-scores by using word vectors [9]. Since word vectors appear to contain information about other relations, we exploit them in our proposed approach.

In this thesis, we introduce new distributed representations for sequences of words between two words, called *substring vectors*. We used it as only a feature set for relation

Table 3.1: Number of features and resources.

System Name	# of Feature Sets	# of External Resources
Baseline	1	0
ECNU-SR-7	5	2
ISI	4	3
FBK_IRST_12VBCA	4	1
UTD	6	5
RMVN	2	1
CDNN	2	1
CR_CDNN	2	1
$\text{RelEmb}_{\text{FULL}}$	3	3
Proposed	**1**	**0**

classification and achieved a sufficiently high accuracy(78.10%.) More importantly, different from existing approaches, our approach do not use any external resources, and has a low number of dimension for feature vectors. The novelty of our approach is that we have proposed a new simple but effective weighting method based on words frequency. The experiments showed that after processing the weighting, the F-scores of the classification results can be improved by 1%–3%.

3.2 Related Work

3.2.1 Learning with Sophisticated Features

Relation classification is one of the most important topics in Natural Language Processing (NLP). A benchmark dataset for semantic relations called SemEval-2010 Task 8, which was used in a contest, picks up nine relations that cover a sufficiently broad range to be of general and practical interest [32]. The approach of Bryan et al., won the relation classification contest [29], uses various types of features, which can be partitioned into eight groups, where five groups are taken from external resources. This shows that the combination of rich features and learning algorithms that are tolerant to high dimensions, such as the linear Support Vector Machine (SVM), is one of the most effective approaches for relation detection. However, the performance of the approach strongly depends on the quality of the designed features and the amount of external resources.

3.2.2 Learning with NNLMs

Bengio (2003) proposed a neural probabilistic language model [4] that is also increasingly used to solve the problems of NLP. With the recent revival of interest in Deep Neural Networks (DNNs), many researchers have concentrated on using deep learning approaches to learn features. Socher (2012) proposed a new Recursive Neural Network (RNN) to learn vectors for relation classification [33]. Motive by RNN, Hashimoto (2013) proposed an explicit weighting of important phrases for the target task. Their experimental results on semantic relation classification show that weighting significantly improves the prediction accuracy of the model [34]. Recent research on relation classification has used a CDNN to extract lexical and sentence level features [8]. Based on CDNN, Sanotos (2015) [35] proposed a pair-wise ranking loss function that makes it easy to reduce the impact of artificial classes. This approach have achieved state-of-the-art results for relation classification.

The above studies showed that using NNLMs improves results more than other existing approaches, and appropriate weighting of features improves the quality of vectors. Unfortunately, because of the large number of dimensions, it is difficult for existing approaches to reduce the computational cost while maintaining prediction accuracy. To solve this problem, we propose low-dimensional feature vectors for relation classification. Our approach effectively alleviates the shortcomings of traditional features as described in the next section.

3.2.3 Comparison with Existing Approaches

In order to express more clearly the novelty and advantage of our work, we compared our proposal with existing approaches from two aspects: one is the novelty and the advantage of our algorithm, and the other one is the simplicity and its advantage as a consequence. We mention here two methods as resent NNLM methods of semantic classification for comparison; Zeng's method [8], which is representative for others using CNNs [9], and Hashimoto's method [36], which is most similar with our method to the best of our knowledge.

First, we describe the novelty and the advantage of our method compared to others. The clearest novelty of our method is that we give a new weighting method to make feature vectors. All method including our method generally divides a sentence into three pieces at first, which are the inside word sequences and the word sequences on the left and the right hand side (i.e., $\texttt{Substr}_2$, $\texttt{Substr}_1$ and $\texttt{Substr}_3$ in Section 3.4.3.) In Hashimoto's method, a vector representation for each inside word are learned by their proposed probabilistic model, where is similar with CBOW but specialized to represent the above segmentation of each

sentence. Then, to make a feature vector, they take an arithmetic mean of all inside word vectors with a certain length of the window. They classify semantic relations by feeding feature vectors to the single-layered softmax classifier. On the other hand, Zeng's method first makes a word vector for each inside word by the well-known embedding methods such as CBOW, and then feeds the sequence of those vectors along with other sophisticated features to the CNN.

Our method is relatively more similar with Hashimoto's method than Zeng's CNN-based method, in terms that both our and their methods averages word vectors corresponding to the inside word sequences. However, our method makes a final feature vector by the specific weighted arithmetic mean, which is one of the main novelties of our method, whereas Hashimoto's method uses the simple arithmetic mean.

As shown in Section 3.6.4, our weighting method improves the result by 1%–3% comparing to the method using the simple arithmetic mean. This improvement is sufficiently considerable comparing to other methods. For example, the specialized word vectors proposed by Hashimoto, which is one of the main novel point of their methods, improves by 1% comparing to the method replacing their specialized word vectors with usual ones obtained by well-known methods like CBOW. In addition, our weighting method is able to be combined with other methods that uses the simple arithmetic mean of word sequences like Hashimoto's.

In the following, we describe the simplicity and its advantage of our approach. One of the main simplicity of our approach is that it has significantly small number of dimensions of input feature vectors, which we call the length of input vectors below. In Zeng's method, the length of input vectors is at least $d(3l + 8)$, where d is the number of dimensions of each word vectors and l is the variable length of the inside word sequence. Since they let $d = 50$ in their paper and l is approximately 17 in average for the used data, the length of input vectors is no less than 2950 approximately. In Hashimoto's method, the length of input vectors is at least $4d(2 + c)$ as they describe, where c is the length of the window each word has. By plugging $d = 100$ and $c = 3$ which are the values they found by tuning, we have that the length of input vectors for their method is approximately 2000. In contrast to those methods, our approach makes input vectors with the length of d, and we find that the results are best when $d = 50$ approximately.

The significantly small length of the input vectors has some obvious advantages. One of them is the low computational cost for both learning and predicting. Unfortunately, we could not find the actual computing time for both Zeng's and Hashimoto's methods. However, if a classifier is fixed, its computational cost for learning and prediction significantly

depends on the length of input vectors in many cases. In fact, the experiments shown in Section 3.6.3 demonstrates that general claim at least for two specific classifiers.

We also point out that no external resources are mandatory in our approach, whereas at least one external resources are used in almost all the existing researches as table 3.1 shows. Generally speaking, non-external-resource approaches reduce the human costs and increase the robustness when applying them to new languages and domains. The experiments show that our approach still achieved a sufficiently high accuracy even though it is non-external-resource and has simple low dimensional feature vectors.

3.3 Distributed Representations of Words

Distributed representations of words in a vector space help learning algorithms to achieve a better performance in NNLM processing tasks, which is also called word embedding or word vectors [3][37]. This idea has been applied to statistical language modeling with considerable success [38][4]. Mikolov et al. proposed two new language models called the Continuous Bag-of-Words (CBOW) model and the Skip-Gram model, both of which are a type of unsupervised NNLM [39][6][31][40]. They also proposed an approach called negative sampling [41], where negative examples that do not exist in data are constructed in order to reduce the problem to an optimization in a discriminative model [42].

In the CBOW model with the negative-sampling approach, a center objective function parameterized by input and output vectors of words is optimized.

Intuitively, this optimization makes the vector of a word w as close as possible to the average of the vectors of words in the context of w with the length of k, while each vector repel each other due to the effect of negative samples. The word vectors obtained above are known to produce good representations that reflect semantics of words.

For relational classifications, since there are multiple words between two objective words, the word-vector representations are still redundant to be considered as input vectors for existing classifiers. Thus we define and construct new efficient vectors for substrings based on word vectors as described in the following section.

3.4 Proposed Method

In this section, we propose a new approach for classification of relations between pairs of nominals. We introduce a simple but effective feature vector called *substring vector*, which is constructed above word vectors. The process for creating the *substring vectors* consists of three steps. Firstly, a vocabulary is constructed from the training text data and then vector

representations of words are learned. Secondly, substrings between pairs of nominals in sentences are extracted and a weight for each word vector is calculated. Finally, after the vectors for each word in a substring are weighted and normalized, the sum of the vectors constitutes the *substring vectors*.

For instance, in the sentence "This chair is made of wood", the words "chair" and "wood" have the `Entity–Origin` relation. The substring specified by these words and to be mapped into $\mathbb{R}^N$ is "is made of". The weight of each word in this substring is different. Words that carry more relation information should be more weighted, while ubiquitous words should not. Our approach defines the weight of each word through its frequency in the text, so that if the word frequently appears in the substring, it has a higher weight score. In the above example, the weight score of "make" and "of" is higher than "is". According to this rule, we can obtain a weight dictionary for words. Then, we can perform weighting and normalizing for the original vector. Using the processed word vector of each word in the substring, we sum them to construct the *substring vectors*. For instance, $v(substring_1) = v(\text{is}) + v(\text{made}) + v(\text{of})$, and $v(substring_1)$ are the corresponding *substring vectors* for "chair" and "wood". The *substring vectors* are a kind of feature for training classifiers for relation classification.

In the following, we provide a detailed description of our approach.

3.4.1 Assumed Input Data

We denote $S_1, \cdots, S_M$ be the sentences in the data and for each i, $S_i = w_{i1} \cdots w_{i|S_i|}$, where w_{ik} represents the k-th word of S_i. We denote the set of all sentences $D = (S_1, \cdots, S_M)$. We assume that each sentence S_i have at most one pair of indices e_{i1} and e_{i2}, where $w_{ie_{i1}}$ and $w_{ie_{i2}}$ are the pair of words to be classified with respect to semantic relations. To escape double subscripts, we let $w(e_{ik})$ denote $w_{ie_{ik}}$ for $k = 1, 2$. If S_i have no such pair of words, we let $e_{i1} = e_{i2} = 0$ and $w(0) = \lambda$, where λ denotes the empty word. Input data is represented by the sequence of triplets $\langle S_1, e_{11}, e_{12}\rangle, \cdots, \langle S_M, e_{M1}, e_{M2}\rangle$. We allow e_{i1} and e_{i2} to be chosen arbitrarily. For instance, every pair of locations of nominals except for stop words appeared in a sentence can be regarded as e_{i1} and e_{i2}. For another instance, if we use data sets like SemEval-2010 Task 8, since each pair of nominals to be classified is marked as indices from the beginning for each sentence S_i, we can use them as e_{i1} and e_{i2}. It is to be noted that this assumption does not include any information about classification for those pairs, since no label except for e_{i1} and e_{i2} is given to S_i.

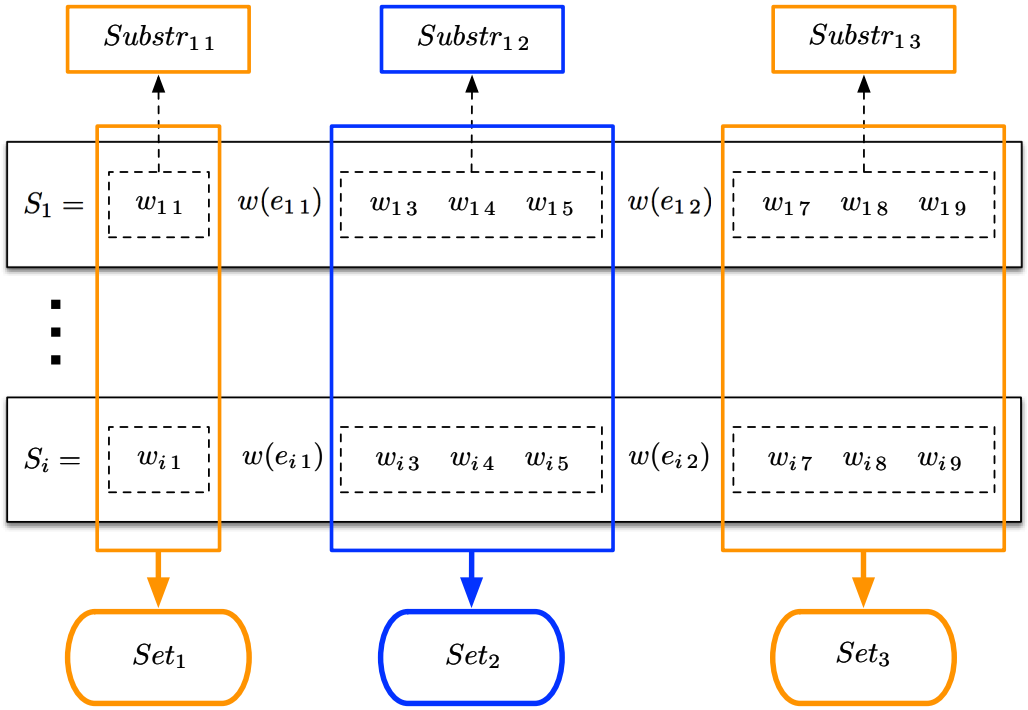

Figure 3.1: Sentence S_{11} is divided into three parts, $\texttt{Substr}_{11}$, $\texttt{Substr}_{12}$, and $\texttt{Substr}_{13}$. $\texttt{Set}_1$, $\texttt{Set}_2$, and $\texttt{Set}_3$ are multisets of substrings.

3.4.2 Learning Vector Representation of Words

Since we construct the *substring vectors* based on word vectors [30], the first step is to obtain word vectors for all words in D. All sentences in D are used as a training data for algorithms, such as the CBOW model with the negative sampling described in Section 3.3. Word vectors are in $\mathbb{R}^N$, where N is arbitrarily chosen but is usually around 10–100. Note that, in this step, although, for each sentence, we have the sequence of word vectors which we believe that have potentially enough information about semantic relations, it is still required to introduce another distributed representation for the sentence itself as described later. Simple concatenation of the word vectors in each sentence is too naive and has too large dimension to be classified well.

3.4.3 Extraction of Substrings from Sentences

Each sentence S_i is divided into three substrings by splitting S_i at e_{i1} and e_{i2}. We denote $\texttt{Substr}_{i1}$, $\texttt{Substr}_{i2}$ and $\texttt{Substr}_{i3}$ be those substrings in order. The substring that we mainly map into the vector space is $\texttt{Substr}_{i2}$, since $\texttt{Substr}_{i2}$ is most informative about the semantic relation between $w(e_{i1})$ and $w(e_{i2})$. For instance, we suppose that the data set D is

consisted of only one sentence i.e., $D = (S_1)$, where

$$S_1 = \text{The}_1\ \text{eye}_2\ \text{works}_3\ \text{using}_4\ \text{the}_5\ \text{retina}_6\ \text{as}_7\ \text{a}_8\ \text{lens}_9\ .,$$

$e_{11} = 2$ and $e_{12} = 6$, i.e., $w(e_{11})$ = "eye" and $w(e_{12})$ = "retina." We denote w_i be the i-th word of S_1. S_1 can be thought as a sentence that describes the relations between e_1 and e_2. Then, we have

$\texttt{Substr}_{11} = (w_{11})$ = ("the"),

$\texttt{Substr}_{12} = (w_{13}, w_{14}, w_{15})$ = ("works", "using", "the"),

$\texttt{Substr}_{13} = (w_{17}, w_{18}, w_{19})$ = ("as", "a", "lens"),

and $\texttt{Substr}_{12}$ looks to explain the relation between "eye" and "retina", and are best among the three. Figure 3.1 shows sentences divided by e_1 and e_2 in our approach. By putting the substring in each sentence of text data together, we can get three multisets of substrings: $\texttt{Set}_1$, $\texttt{Set}_2$ and $\texttt{Set}_3$. e_1 and e_2 might appear continuously in one sentence, that means exist sentence without $\texttt{Substr}_2$. This sentence will be treated as invalid data, and not within the scope of our approach. But $\texttt{Substr}_1$ and $\texttt{Substr}_3$ as the empty is allowed.

3.4.4 Constructing Distributed Representation of Substrings

After word vectors are leant and all sentences are divided into substrings, we make *substring vectors*. The process of construction is shown in Figure 3.2. First, we create the weight dictionary based on word frequencies. Then, we use this dictionary for weighting and normalization of each words in substrings and obtain *substring vectors* by averaging them.

3.4.4.1 Normalized Weight of Words in Substrings

To construct a reasonable representation for a substring from word vectors, we define weights for each words which represent degrees of importance of relations between pairs of nominals. For instance, suppose that a data set D includes only one sentence S_1, i.e., $D = (S_1)$. Among "works", "using" and "the", which are elements of $\texttt{Substr}_{12}$, "the" appears in both $\texttt{Substr}_{11}$ and $\texttt{Substr}_{12}$ while "works" and "using" appear only in $\texttt{Substr}_{12}$. Thus, we observe that, compared to "the", "works" and "using" are more informative for the semantic relation between "eye" and "retina." In other words, if a word w appears mainly in $\texttt{Set}_2$, we believe that w often plays a role of representing some semantic relation.

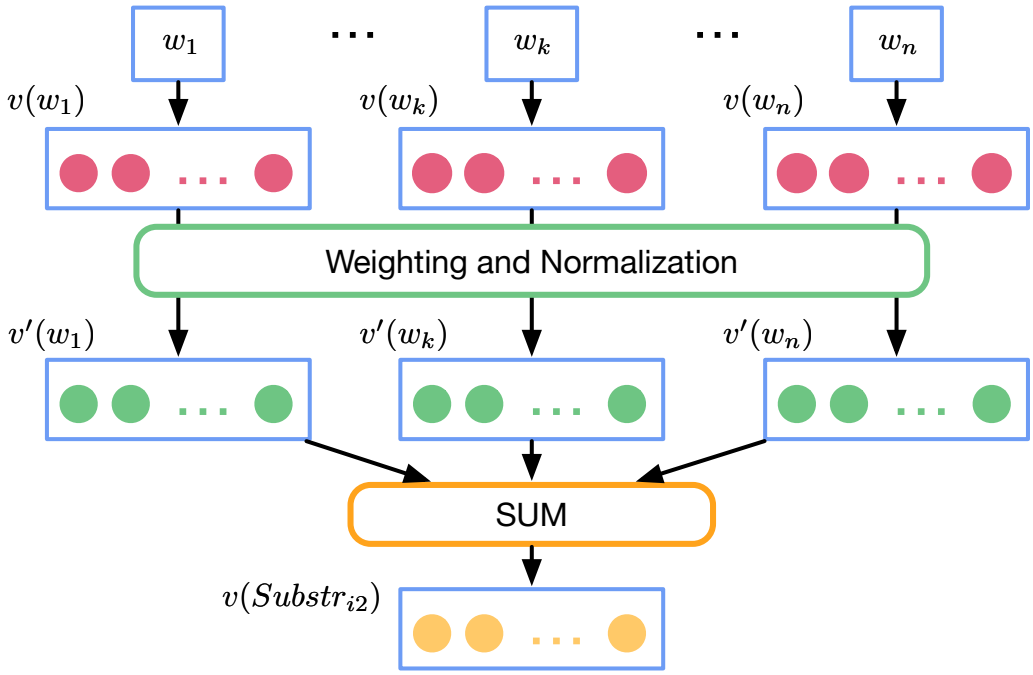

Figure 3.2: Construction process of substring vectors

For a word w and a substring s, we define $\texttt{Cnt}(w, s)$ as the number of occurrences of w in s. In addition, for a multiset of substrings $\mathcal{S}$, we denote $\texttt{Cnt}(w, \mathcal{S}) = \sum_{s \in \mathcal{S}} \texttt{Cnt}(w, s)$. The weight for a word w is defined as follows:

$$a(w) = \frac{\texttt{Cnt}(w, \texttt{Set}_2)}{\texttt{Cnt}(w, \texttt{Set}_1) + \texttt{Cnt}(w, \texttt{Set}_2) + \texttt{Cnt}(w, \texttt{Set}_3)}. \tag{3.1}$$

In order to prevent that the length of each *substring vector* is too large or too small, or to have the center of gravity for weighted word vectors, we normalize the weights in Eq. 3.1 with respect to each substring $\texttt{Substr}_{ij}$:

$$\overline{a}_{ij} = \frac{a(w_{ij})}{\sum_{k=e_{i1}+1}^{e_{i2}-1} a(w_{ik})}. \tag{3.2}$$

3.4.4.2 Vector Representation of Substring

Using the normalized weights and the word vectors, we define the *substring vectors* $v(\texttt{Substr}_{i2})$ for each substring $\texttt{Substr}_{i2}$ as follows:

$$v(\texttt{Substr}_{i2}) = \sum_{k=e_{i1}+1}^{e_{i2}-1} \overline{a}_{ij} v(w_{ij}). \tag{3.3}$$

where $v(w)$ is the word vector of w. shown in Figure 3.2,

In many language models, word vectors of a given sentence or substring are simply concatenated and treated as an input vector in their lowest layer. However, since this approach increases the dimension of input vectors, it courses high computational cost. Furthermore, since the number of words in a sentence or substring can be changed, it is also difficult to unify the number of dimensions. Based on the above two points, our approach proposes a simple way to make the *substring vectors*.

In previous researches, it is known that making input vectors including external knowledge such as WordNet increases the prediction accuracy. In this thesis, to verify effectiveness of our approach, we use only *substring vectors* as input vectors for classifiers.

3.4.5 Multiclass Classifiers

In recent research, the dimension of feature vectors tends to be extremely large since including much information as far as effective is believed to advocate high accuracy classification. In those cases, leaning algorithms must have simple structures such as linear and at most small-order polynomial models due to the problems of computational cost and overfitting. On the other hand, in our approach, we focus on exploring a future space which is enough low-dimensional and smart so that words are mapped keeping their semantic relations. In stead, we use rather non-linear and flexible classifier such as SVM with gaussian kernels, where it is hard to apply to extremely high-dimensional data sets.

3.5 Implementation

We implemented the proposed method in Java, and used two external open-source softwares. In semantic relation classification, we choose the same text data as other similar researches, which includes two parts, train and test datasets. We calculated their substring vectors as feature vectors for each part, and applied classifiers to obtain classification results.

3.5.1 Dataset

To evaluate the performance of our proposed method, we used the SemEval-2010 Task 8 dataset [32]. The dataset is freely available and contains 10,717 annotated examples, including 8,000 training instances and 2,717 test instances. It distinguishes nine semantic directed relations, such as `Entity–Origin`, `Component–Whole`, and `Cause–Effect`, shown in Table 3.2 [32]. In addition, it has another special undirected relation called `Other`.

Table 3.2: The details of the relations in SemEval-2010 Task 8 dataset

Relation	Definition	Example
Cause-Effect (CE)	An event or object leads to an effect.	those cancers were caused by radiation exposures
Instrument-Agency (IA)	An agent uses an instrument.	phone operator
Product-Producer (PP)	A producer causes a product to exist.	a factory manufactures suits
Content-Container (CC)	An object is physically stored in a delineated area of space.	a bottle full of honey was weighed
Entity-Origin (EO)	An entity is coming or is derived from an origin (e.g., position or material).	letters from foreign countries
Entity-Destination (ED)	An entity is moving towards a destination.	the boy went to bed
Component-Whole (CW)	An object is a component of a larger whole.	my apartment has a large kitchen
Member-Collection (MC)	A member forms a nonfunctional part of a collection.	there are many trees in the forest
Message-Topic (MT)	A message, written or spoken, is about a topic.	the lecture was about semantics

In this dataset, as shown in Figure 3.3, each training sentence has one pair of nominals tagged with <e1> and <e2> and specifies one semantic relation. Note that semantic relations have a direction. For instance, although both (2) and (1) in Figure 3.3 have the relationship called `Member–Collection`, these two instances cannot be classified into the same category when we attempt to learn, because `Member–Collection` (e2,e1) and `Member–Collection` (e1,e2) should be distinguished.

3.5.2 External Tools and Hyperparameters

We used two external software programs for the implementation. The first one is called Word2vec[1] and is for learning word vectors from sentences based on the CBOW model. Before learning the word vector, we had to clean the data; for instance, we removed tags in given sentences, unnecessary symbols, and substrings in parentheses. The parameters for word2vec were set as follows. We set the embedding dimensionality to {20, 40, 60, 80, 100, 200, 300, 400, 500} , the $window = 5$, $sample = 1e-4$, $iter = 15$, $cbow = 1$, and

[1]https://code.google.com/p/word2vec

(1)	"It describes a method for loading a horizontal <e1>stack</e1> of containers into a <e2>carton</e2>." : `Entity–Destination` (e1,e2)
(2)	"There is a <e1>nest</e1> of <e2>rabbits</e2> up in the loft." : `Member–Collection` (e2,e1)
(3)	"This <e1>surgeon</e1> is part of the study <e2>group</e2>." : `Member–Collection` (e1,e2)

Figure 3.3: Examples of sentences and their semantic relations labels.

the learning rate be the default value. We extracted a corpus from the SemEval-2010 Task 8 dataset. It includes 10,717 sentences and 184,877 words (1.1 MB.) Then the word2vec was trained by this corpus.

The second software is called Weka[2] and is for the classification. After the substring vectors as feature vectors were constructed, we applied three multi-class classifiers for them: random forest (RF) [43][44], SVM [7] with the Gaussian radial basis function kernel (SVM-RBF), and the linear SVM (SVM-LN). We set almost all the hyperparameters as default, except for the number of trees for RF, and the penalty term for incorrect Classification of SVM-RBF. Those two hyperparameters are searched by cross validation. As a result, for the parameter settings of RF, we let the number of trees be 120. For those of SVMs, we let the penalty term for incorrect Classification of SVM-RBF be 60. In Section 3.6.7, We show the sensitivity of hyperparameters and how we determined them.

3.6 Experiments

In this section, we conducted five sets of experiments for evaluating our approach. The results of experiments demonstrated that our approach yielded better classification results(F-score: 77.18% and 78.10%) than most of sophisticated features approaches, however there was still a gap in the state-of-the-art methods which used NNLMs (almost all F-score above 80%.) Crucially, unlike other existing approaches, our method did not use any external resources and the feature vectors had a sufficiently low-dimensional(40–60). In addition, we verified the validity of the proposed weighting method and dimension reduction method through experiments.

[2]http://www.cs.waikato.ac.nz/ml/weka

3.6.1 Measure of Evaluation

We learned from the training data and obtained F-scores from the test data for 10 relations including `Other`. The average of the F-scores for nine relations excluding `Other` is called the macro-averaged F-score. In addition, we removed the instances of `Other` from the training and test data and obtained the average of the F-scores for nine relations, which is called micro-averaged F-score. To compare results of our proposed method in Section 3.6.2, we adopted the macro-averaged F-score as a measure of the prediction accuracy that is same as previous studies. However, we adopted the micro-averaged F-score in other experiments (Section 3.6.3–Section 3.6.6) because the classification results will be more stable when excluding occurrences of `Other`.

3.6.2 Comparison of the Proposed Method with Existing Methods

To evaluate the performance of our proposed method, we compared six methods with our method, as shown in Table 3.3. The first four were the best of the existing approaches that are not NNLMs and the following two were proposed in current studies using NNLMs. The Table 3.4 shows F-scores as prediction accuracies for training sets consisting of 1000, 2000, 4000, and 8000 sentences (TD1–TD4). Since the initial vectors of Word2vec are random, the learnt vectors have also a certain range of changes. Thus, we took the average values of 10 executions to avoid instability in our experimental results. In addition, to compare our results with those obtained in previous studies, we adopted the macro-averaged F-score as a measure of prediction accuracy.

The F-scores of our proposed methods used the weighted method, and used CBOW model for learning the word vector representations. At the beginning of experiments, we used two models, the CBOW and the Skip-gram. However we found that the performance of CBOW was better than Skip-gram in F-scores (0.9%–1.5%.) The number of dimensions of RF is 60 and SVM-RBF is 40. The results were obtained without the using of PCA and ICA [46][47].

As shown in Table 3.4, in spite of the very simple feature set, the results of the proposed methods were comparable with other methods in terms of prediction accuracy. The table shown that the existing approaches use various kind of features, including external data sources, to produce comparable results. While NNLM methods (RMVM and CDNN) deviated from this tendency of existing approaches, they still use external data sources, such as WordNet. External data sources can provide abundant data that has been classified. For example, there are two semantic relationships “has-part” and “part-of” in WordNet, as

Table 3.3: Classifier, feature sets, and resources used for relation classification

System	Classifier	Feature sets	Resource used
Baseline [32]	Naive Bayes	local context of 2 words only	-
ECNU-SR-7 [27]	SVM	stem, POS, syntactic patterns, hyponymy and meronymy relations	Word Net and syntax
ISI [45]	Maximum Entropy	a noun compound relation system, various feature related to capitalization, affixes and closed-class words	Word Net , Roget's Thesaurus and Google n-grams
FBK_IRST _12VBCA [28]	SVM	3-word window context features (word form, part of speech, or orthography) + Cyc; parameter estimation by optimization on training set and verbs	Cyc
UTD [29]	SVM and two-step classification	context words, hypernyms, POS, dependencies, distance, semantic roles, Levin classes, para-phrases	WordNet, syntax, Google n-grams, PropBank/NomBank and Levin classes
RMVM [33]	MVRNN	POS and NER	WordNet
CDNN [8]	softmax	word pairs, words around word pairs	WordNet
CR_CDNN [35]	softmax	word embeddings, word position embeddings	Wikipedia
$\text{RelEmb}_{\text{FULL}}$ [36]	softmax	embeddings, dependency paths, WordNet, NE	WordNet,Named Entity tags,Wikipedia
Proposed	**RF, SVM-RBF**	**substring vectors**	-

shown in Table 1.1. The data contained in these two relationships can be used as a correct answer for "Member-Collection(MC)". As we have mentioned earlier, this behavior

Table 3.4: F-score of all systems for the test dataset as a function of training data: TD1=1000, TD2=2000, TD3=4000, and TD4=8000 training examples.

System	TD1	TD2	TD3	TD4	Best Cat	Worst Cat
Baseline	33.04	42.41	50.89	57.52	MC (75.1)	IA (28.0)
ECNU-SR-7	58.67	58.87	72.79	75.21	CE (86.1)	IA (61.8)
ISI	66.68	71.01	75.51	77.57	CE (87.6)	IA (61.5)
FBK_IRST_12VBCA	63.61	70.20	73.40	77.62	ED (86.5)	IA (67.3)
UTD	73.08	77.02	79.93	82.19	CE (89.6)	IA (68.5)
RMVM	-	-	-	82.4	-	-
CDNN	-	-	-	82.7	-	-
CR_CDNN	-	-	-	84.1	-	-
$\text{RelEmb}_{\text{FULL}}$	-	-	-	83.5	-	-
Proposed (RF)	**69.17**	**72.27**	**75.07**	**77.18**	CE (92.50)	CW (67.00)
Proposed (SVM-RBF)	**70.84**	**73.11**	**76.38**	**78.10**	CE (92.90)	CW (65.70)

of using external data increases the complexity of the feature set which resulting in an increased processing time. Thus, it is difficult to simplify the complexity of the features and improve classification results simultaneously. The proposed method used only one kind of feature without any external data source. As a result, the computational cost for learning and classifying is also sufficiently small. The proposed method appeared to successfully achieve relatively simple features, small computational cost, and high classification accuracy simultaneously.

3.6.3 Comparison of Scores and Computing Time in Different Dimensions

Figure 3.4 and 3.5 shown the F-scores and computing time for the first set of experiments. If the dimension of the *substring vectors* is less than approximately 400, the nonlinear classifiers obtained better classification results than the linear one. However, when the dimension is greater than approximately 400, the linear classifier had better performance than the others. This phenomenon appears to be due to the overfitting caused by the high degree of freedom that the SVM-RBF is as a classifier. This implied that a sufficiently

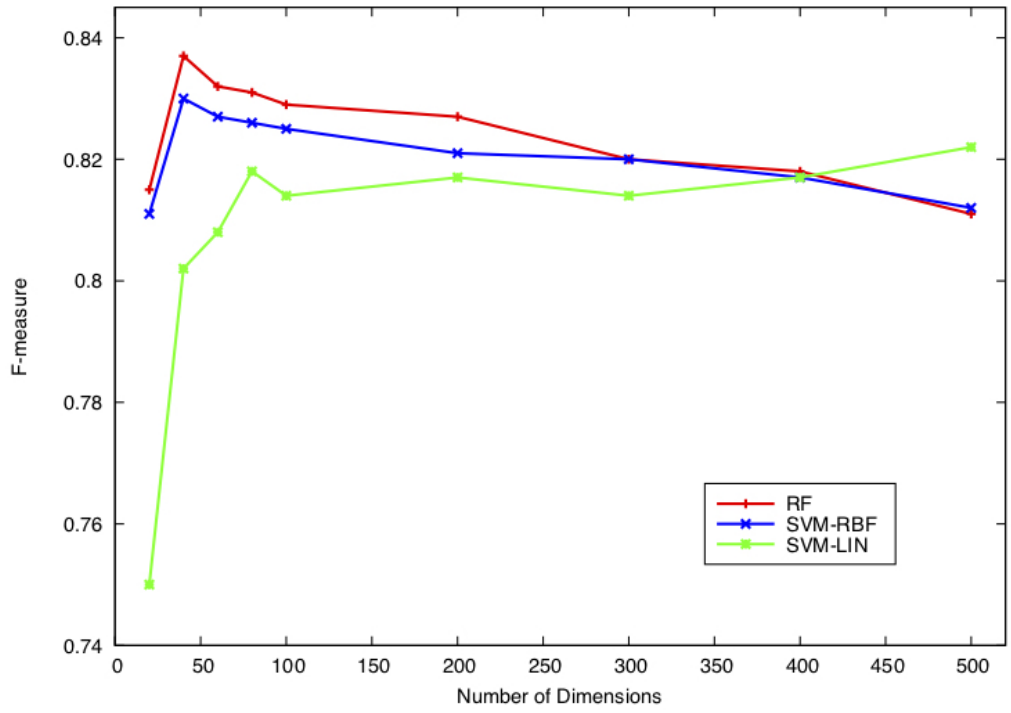

Figure 3.4: F-scores for each classifier as a function of the dimension of the substring vectors.

low-dimensional representation of the data is required in order to use nonlinear classifiers efficiently. In the second experiment, we obtained the best classification results using nonlinear classifiers for approximately 50 dimensions of the *substring vectors*, and the RF obtained better performance than the other classifiers with respect to the computing time.

3.6.4 Effect of the Proposed Weighting Method

In this set of experiment, as shown in Figure 3.6, after processing the weighting, the F-scores of the classification results can be improved by 1%–3%. We ensured that our weighting method is effective by comparing the weighted and non-weighted substring vectors as feature vectors of RF. This is because words that appear frequently in substrings between pairs of nominals to be classified (e_{i1} and e_{i2}) are expected to carry much information about semantic relations. Besides, we have tried three other kinds of weighting methods as shown in Table 3.5. Our approach without weighting to be used as a baseline, and W4 represents the weighting method we adopted in our approach as described in Section 3.4.4.

Here, we explain the details of weighting methods W1, W2 and W3. In all weighting methods, we define a corpus C and a word w in a sentence $S_w \in C$. The weight score a_1 of method W1 for the word w is defined as follows: $a_1(w) = \log(|C|/\texttt{Cnt}(w, C_{other}))$.

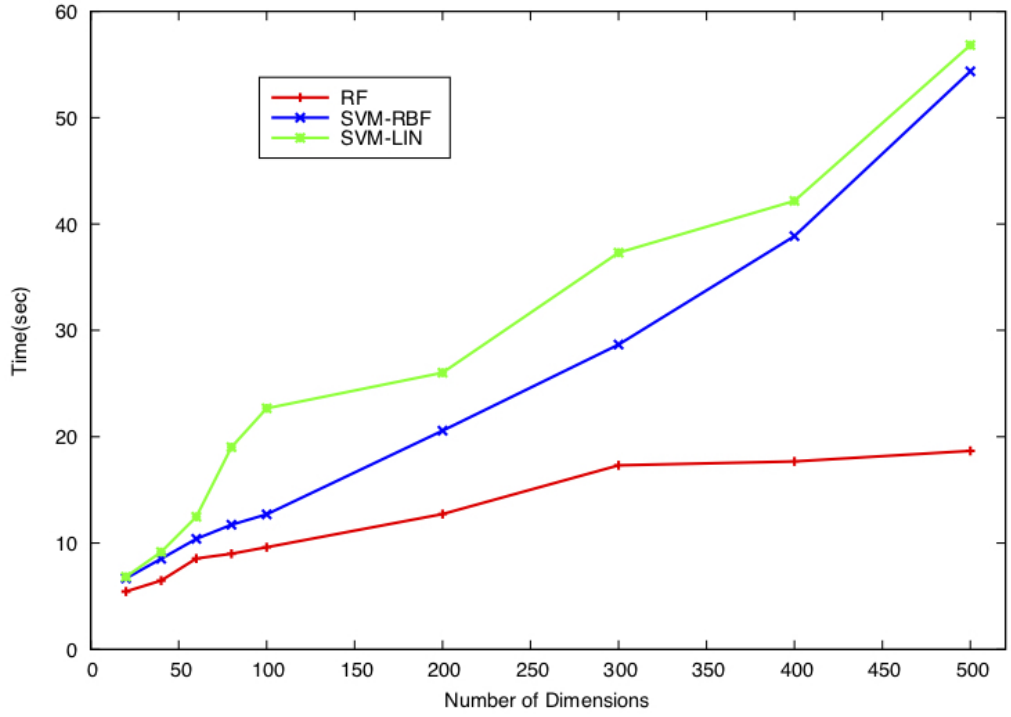

Figure 3.5: Computing time for each classifier as a function of the dimension of the substring vectors.

The intuitive meaning of W1 is that in corpus C, if the word w frequently appears in other sentence $C_{other} = C - \{S_w\}$, it has a lower weight score. The method W2 the same as W1 except counting extent for the word w. The weight score a_2 of W2 is defined as follows: $a_2(w) = \log(|C|/\mathtt{Cnt}(w, M_{other}))$, where M_other is the same set as $\mathtt{Set}_2$ except for $\mathtt{Substr}_2$ of S_w. The meaning of C_{other} and M_{other} as shown in Figure 3.7. In the method W3, the weight of a word w depends on the target label of semantic relation. We divide C into $\{C_{label1}, ..., C_{label9}\}$ according to their target label of semantic relation. Let $\mathtt{Cnt}(w, C_{labeli})$ be the number of occurrences that w appears in C_{labeli}. The weight score $a3$ of method W3 for a word w is defined as follows: $a_3(w) = |C|/\mathtt{Cnt}(w, C_{labeli})$.

Through the experimental results, we can see that the method W1, W2 and W3 shows the worst performance. We considered the case where there are not enough training data to obtain the weights. Therefore, the weighting method W4 is most effective from a small amount of learning data.

3.6.5 Number of Dimensions and Degree of Freedom for Classifiers

The results of the fourth set of experiments are shown in Figure 3.8. We ensured that overfitting occurs more easily for a higher degree of freedom for the classifiers, which is a

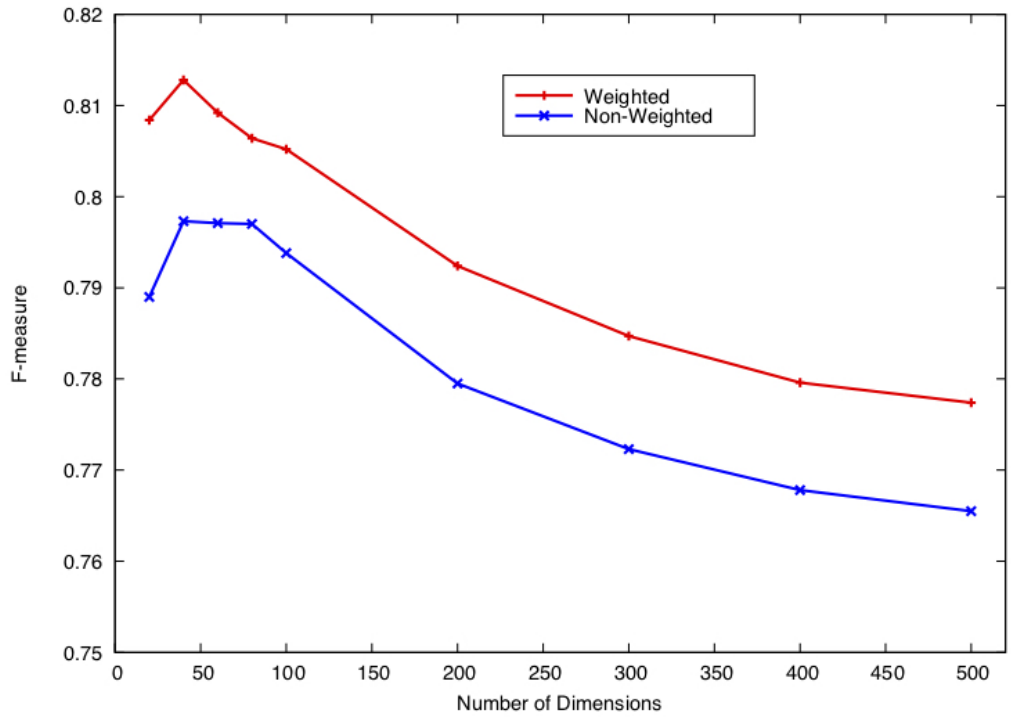

Figure 3.6: Comparison of the effect of *substring vectors* (weighted average) with the simple average of word vectors as feature vectors. In this experiment, we measure scores using 10-fold cross-validation (CV) in the training data for stability.

Table 3.5: F-score of weighting methods ($d = 40$).

Weighting Method	F
NonW(baseline)	79.73
W1	51.4
W2	50.9
W3	52.1
W4	**81.28**

similar result obtained in the first set of experiments. We had the best combination when the degree of the kernel function is not one and the number of dimensions is approximately 40–60.

3.6.6 Effect of Dimension Reduction using PCA and ICA

The purpose of the last two sets of experiments is to examine whether the decrease of the accuracy is caused by the classifiers or the word embedding methods. After we constructed word vectors in 500 dimensions, we reduced the dimensions of the word vectors

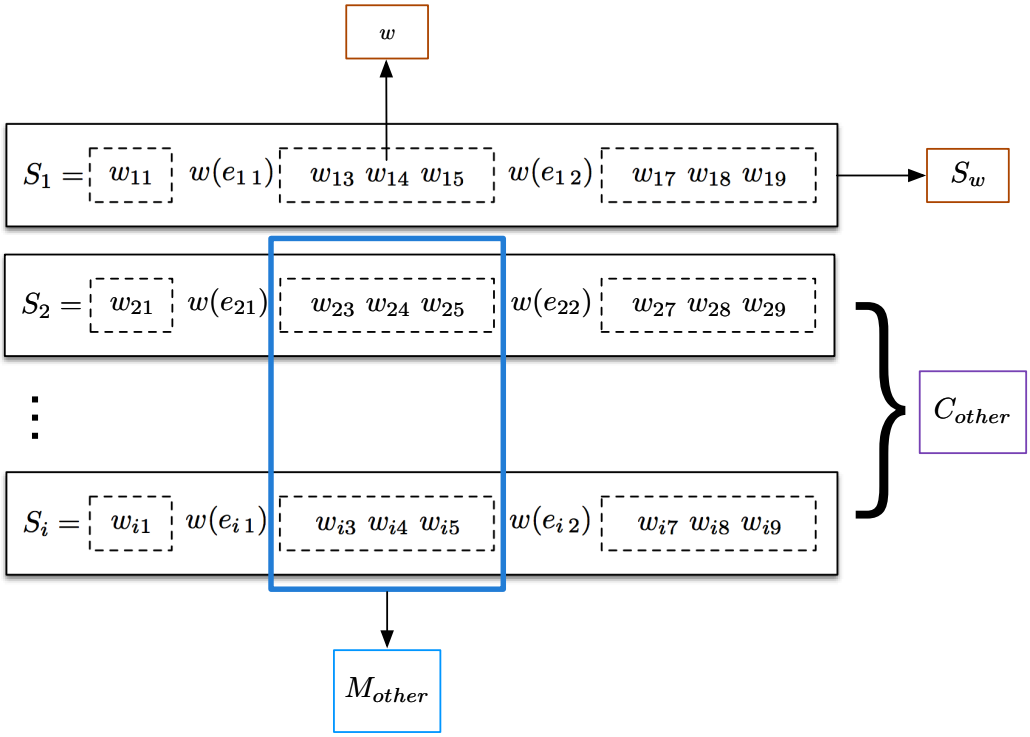

Figure 3.7: In corpus C, C_{other} is a set of sentences and M_{other} is a set of `Substr2`.

by Principal Component Analysis (PCA) and Independent Component Analysis (ICA). The eigenvalues for PCA are shown in Figure 3.9. The figure shown that almost all the word vectors are scattered approximately in a 100 dimensional subspace. We extracted several groups of reconstructed vectors in different dimensions. The classification results of each group are shown in Figure 3.10. The F-score of our method used word vectors in 50 dimensions with PCA is almost same or a little lower than that in 50 dimensions without PCA. The F-score of the original 500 dimensions is 0.811 as a baseline for comparison. If the dimension of reconstructed vectors is less than 300, the processed data obtained better classification results than the original 500 dimensions. When the dimension is greater than 300, the F-score decrease. We observed that even if PCA does not reduce the number of dimensions, that is, when the dimension after PCA is still 500, the score get worse than the original one. This is because PCA transforms the coordinate and changes the distances among embedded words. We also observed that whichever PCA or the word-embedding method we use for reducing the number of dimensions, the score is maximized at 40–50. That fact indicated that the decrease of the accuracy is mainly caused by classifiers or the curse of dimensions.

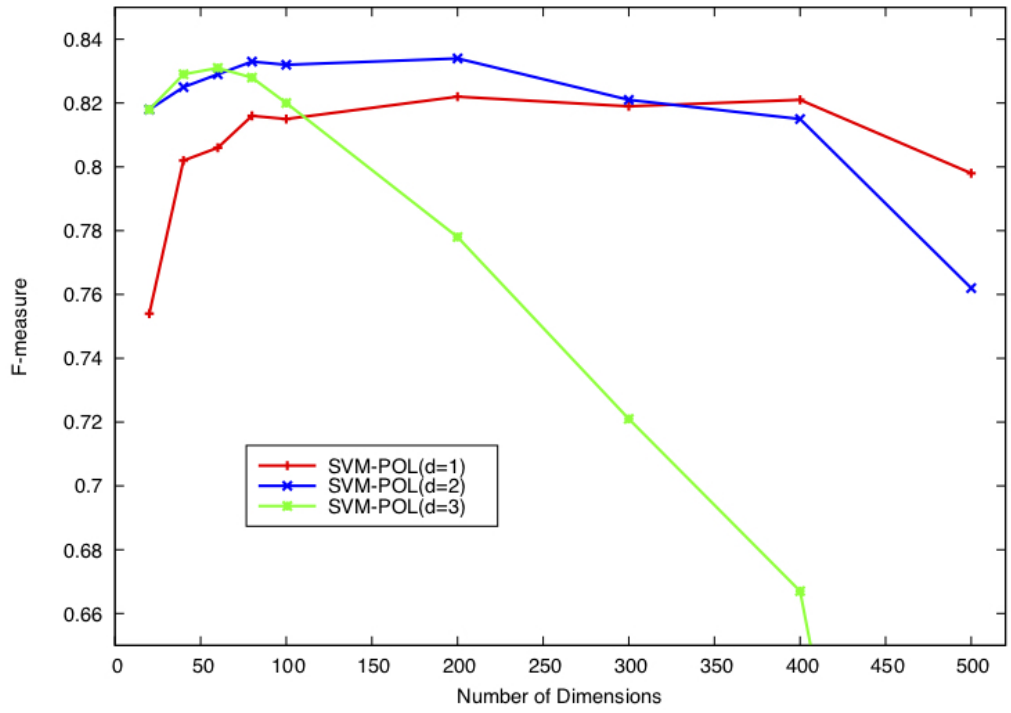

Figure 3.8: F-scores for each degree of the polynomial function kernel SVM as a function of the dimension of the substring vectors.

3.6.7 Determination of Hyperparameters

We show the adjustment of the random forest classifier parameters has a strong influence on the classification results in Figure 3.11. Since we do not optimize parameters other than the number of trees in the case of RF, if the parameters of the classifiers are fully optimized, we may achieve results that are a little better than those presented in this thesis. We observed that a similar phenomenon occurs on RBF classifier when adjusting the penalty parameter.

3.7 Conclusion

In this chapter, we proposed a new distributed representations–*substring vector*, and used it as a feature set for relation classification. Through this simple vector representation, we successfully extracted information about semantic relations between pairs of nominals. With almost no optimizing the parameters of the classifier and without using any external resources, our approach yielded comparable classification results with most of sophisticated features approaches. We also shown that the proposed weighting method of *substring vectors* can improve the the results of relation classification by 1%–3% compared to the non-weighting method. Although we used nonlinear classifiers such as an RF and

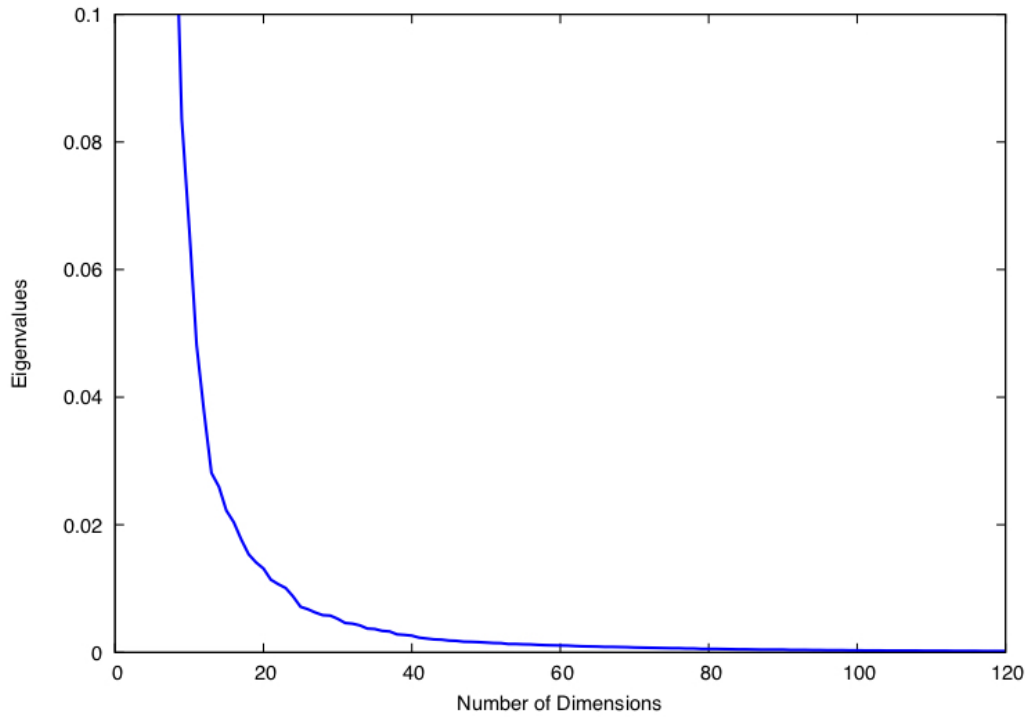

Figure 3.9: Eigenvalues as a function of the number of dimensions for the PCA of the original word vectors.

SVM-RBF, we hope that it may be improved by using other recent learning algorithms used in NNLMs. In addition, the length of the input feature vectors is significantly small compared to that of other existing methods. For instance, when we extract semantic relations from massive amount of unlabeled text data, it is preferable that the size of the set of features is sufficiently small so that data is processed in low-computational cost. The *substring vector* is applicable as an effective feature and is able to be combined with existing features.

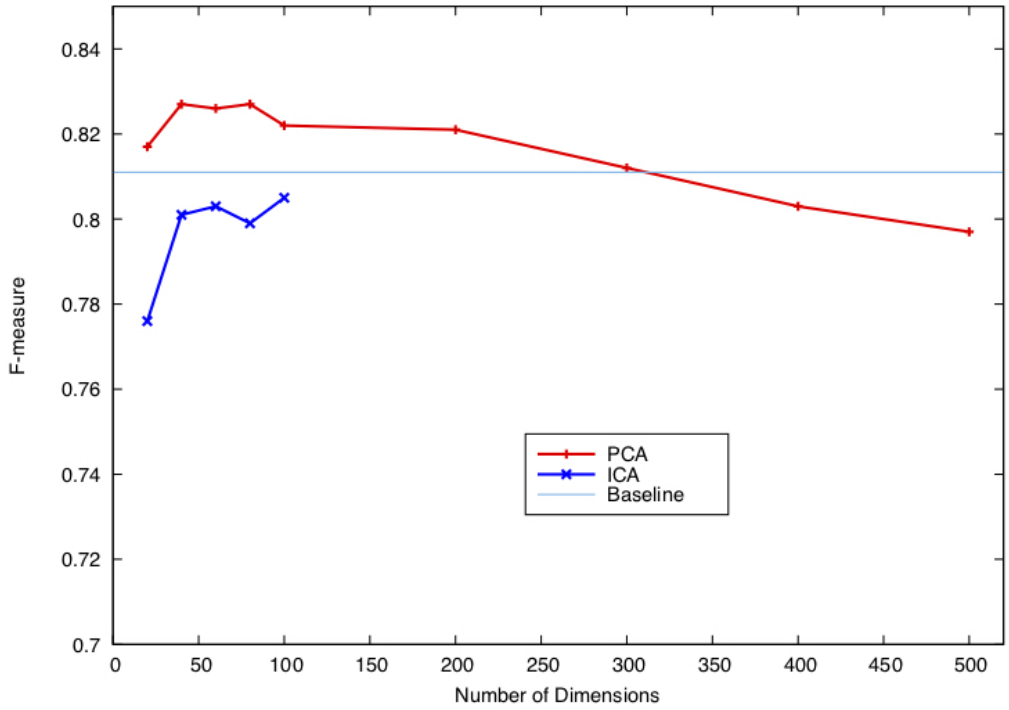

Figure 3.10: F-scores for each dimension of the transformation word vectors. The classifier is the RF, and the baseline is F-scores of the original 500 dimensions.

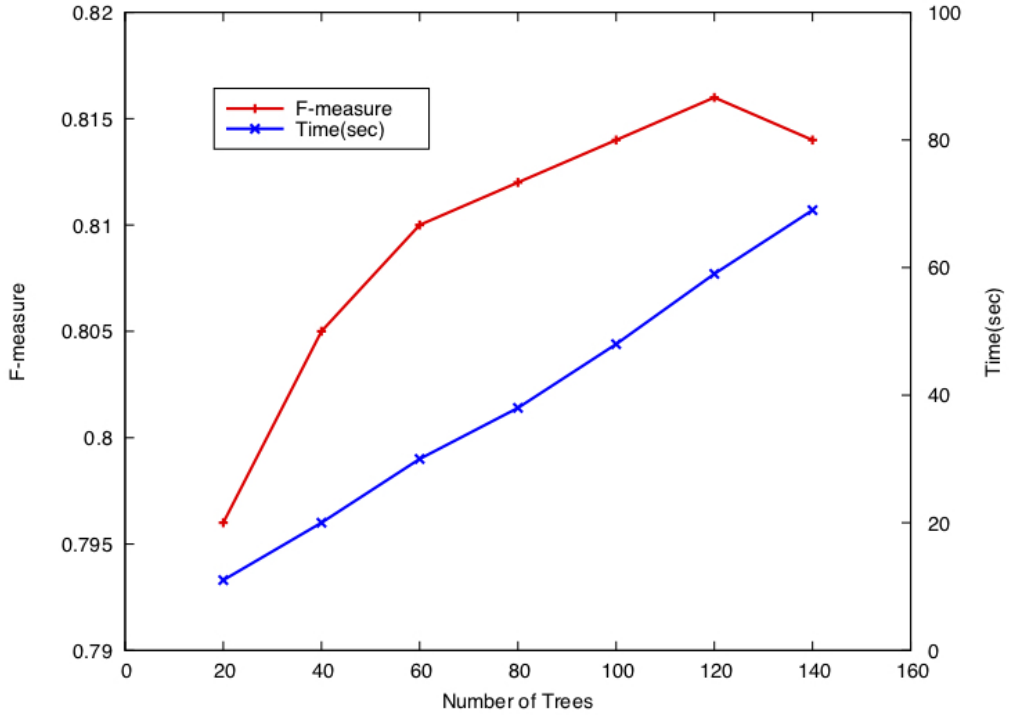

Figure 3.11: F-score and learning time of RF as a function of a parameter of RF (the number of learnt trees). In this experiment, we measure scores using 10-fold cross-validation (CV) in the training data for stability.

Chapter 4

Label Classification of Educational Data through Distributed Representations of Sentences

4.1 Introduction

4.1.1 Analysis on collaborative process

The largest problem in Computer Supported Collaborative Learning (CSCL) study as of now is to analyze its social and cognitive processes in detail in order to clarify what kinds of knowledge and meanings were shared within a group as well as how and by what arguments knowledge construct was achieved. In addition, it is also required to develop CSCL system and tools with scaffolding function which may activate collaborative process by utilizing such knowledge.

A quantitative analysis alone is not sufficient at all to analyze the collaborative process, however, it is associated with a shift to qualitative analysis. Because main data for analysis include contributions over chatting, images and voices on tools such as Skype, and various outputs prepared in the course of collaborative learning, it is totally inadequate to perform just quantitative analysis in order to analyze such data [48][49][50][51].

As these studies often result in in-depth case study, however, they have a downside that it is not easy at all to derive guidelines with generality which are applicable also to other contexts. Therefore, studies have been conducted in recent years based on an approach of verbal analysis in which labeling for appropriately representing properties (hereinafter referred to as coding) is performed to each contribution in linguistic data of certain volume generated over the collaborative learning from perspectives of linguistics and collaborative learning activities [52]. On the other hand, an advantage of the approach is its capability of

quantitative processing for significantly large scale data while keeping qualitative perspective. However, it is a task requiring significant time and labor to perform coding manually and it is expected to become impossible to perform coding manually in a case that data becomes further bigger in size.

In our research project, we have achieved certain results in a series of previous studies reported in eLmL 2017 and the like using deep learning technique for automatic coding of vast amount of collaborative learning data [53][54][55]. In this chapter, while verification is performed for accuracy of the automatic coding based on deep learning technique similarly to eLmL 2017, supervised data has been constructed by conducting coding manually depending on adopted multi-dimensional coding scheme in order to newly recognize collaborative learning process in a more multilateral and comprehensive manner. Based on deep learning by using the data, its accuracy is inspected. In the Chapter 2 and Chapter 3, we proposed two methods for relation classification through distributed representation, but in this chapter, our work was to apply the proposed method to authentic educational settings.

4.1.2 Objective of study

The final goal of our research project is to implement support at actual learning and educational settings such as real time monitoring of collaborative process and scaffolding for inactive groups based on analyses of large scale collaborative learning data as mentioned above. As further development of our previous study, a technique for automatizing coding of chat data is developed based on a multi-dimensional coding scheme capable of expressing collaborative learning process more comprehensively and its accuracy is verified in this chapter.

Specifically, after newly performing coding manually for substantial amount of chat data which was used in the previous studies, a part of it is learned as training data by deep learning methods and then automatic coding is conducted for the test data, And evaluated its accuracy We investigated what kinds of knowledge can be obtained by performing automatic coding on chat data at new cooperative learning which is different from these experimental data.

4.2 Previous Studies

In the previous study, Shibata et al. [53] proposed a coding label consisting of 16 labels (Table 4.1) as a scheme for cooperative learning analysis. Learning was conducted using

depth learning for previous research. The prediction accuracy is achieved with relatively high accuracy. The outline is as follows.

Table 4.1: List of labels

Label	Meaning of label	Contribution example
Agreement	Affirmative reply	I think that's good
Proposal	Conveying opinion, or yes / no question	How about five of us here make the submission?
Question	Other than yes / no question	What shall we do with the title?
Report	Reporting own status	I corrected the complicated one
Greeting	Greeting to other members	I'm looking forward to working with you
Reply	Other replies	It looks that way!
Outside Comments	Contribution on matters other than assignment contents / Opinions on systems and such	My contribution is disappearing already; so fast! / A bug
Confirmation	Confirm the assignment and how to proceed	Would you like to submit it now?
Gratitude	Gratitude to other members	Thanks!
Complaint	Dissatisfactions towards assignments or systems	I must say the theme isn't great
Noise	Contribution that does not make sense	?meet? day???
Request	Requesting somebody to do some task	Can either of you reply?
Correction	Correcting past contribution	Sorry , I meant children
Disagreement	Negative reply	I think 30 minute is too long
Switchover	A contribution to change event being handled, such as moving on to the next assignment	Shall we give it a try?
Joke	Joke to other members	You should, like, learn it physically? :)

4.2.1 Conversation Dataset

Conversation dataset for the study conducted is based on conversations among students obtained from chat function within the system performing online collaborative learning by using CSCL originally developed by the authors for lectures in the university [56]. By the way, we will add that this data is also used in the research of this study. Usage situation of CSCL as the source of the dataset is shown in Table 4.2. Since students participated

in multiple classes, number of participant students is less than the number obtained by multiplying number of groups and that of group members.

Table 4.2: Contributions data used in this study

Number of Lectures	7 Lectures
Member of Groups	3-4 people
Learning Time	45-90 mintutes
Number of Groups	202 groups
Number of Students	426 students
Dataset	11504 contributions

4.2.2 Coding Scheme

According to a manual for coding prepared by the authors, a label was assigned to each contribution of chat. Two coders each coded about all chats and one label was given for one speech of chat. We examined the result of coincidence or mismatch of these codes with the authors and found that there was a blurred code by the coder, so we recoded a part of the code. Any of the 16 types of labels as shown in Table 4.1 was assigned. The ratio of each label is shown in Figure 4.1.

4.2.3 Automatic Coding Approach Based on Deep Learning

In the previous study, we adopted three types of Deep Neural Network (DNN) structures: 1) Convolutional Neural Networks (CNN), 2) Long-Short Term Memory (LSTM) and 3) Sequence to Sequence (Seq2Seq). Of the three models, Seq2Seq model is a deep neural network consisting of two LSTM units called encoder and decoder, and learning of classification problem and sentence generation is performed by entering pairs of strings of words to each part [57][58]. For example, the pair corresponds to a sentence in certain language and its translated sentence in case of translation system as well as to question sentence and response sentence in case of question and answer system, respectively.

In addition, a model based on Support Vector Machine (SVM), which is a traditional machine learning approach is used as a baseline. Accuracy of each model is verified by comparing automatic coding concordance rate and Kappa coefficient. About technology and experiment results in detail for each classification model in existing literatures of the authors [53][54][55].

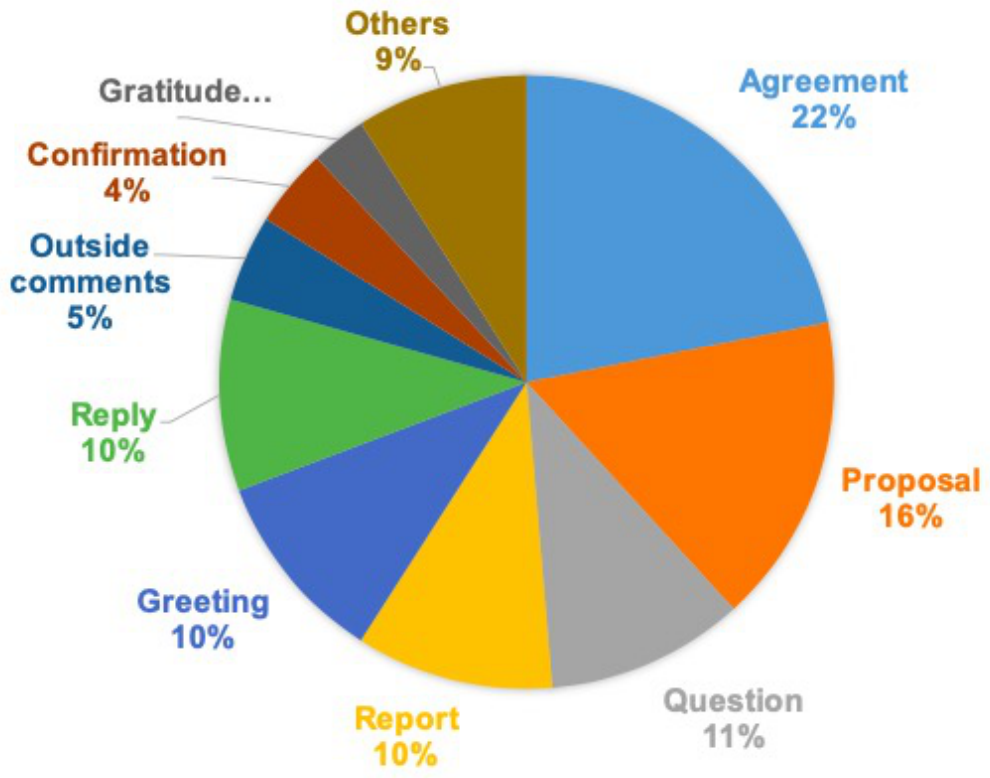

Figure 4.1: Ratio of each conversational coding labels

4.2.4 Experiment and Assessment

4.2.4.1 Outline of experiment

For the data set with manually prepared coding labels as described above, we compared the prediction accuracy of automatic coding for each model. With separation of sentences into morpheme using MeCab conducted at first as a preprocessing of data, words with low use frequency were substituted by "unknown". Subsequently, just 8,015 contributions were extracted and 90% and 10% of them were sorted into data for training and test, respectively. Naive Bayes, Linear SVM, and SVM based on RBF Kernel were applied as baseline approaches.

4.2.4.2 Experiment Results

Table 4.3 shows prediction accuracy (concordance rate) of models proposed in the previous study and those adopted as baseline for test data. The concordance rate here refers to a proportion that manually assigned label conforms with predicted label output by a model. It is proved, as Table 4.3 shows, that accuracy of the proposed model's result is higher than

that of baseline model. Among the three models as described above, it is found that there is almost no difference in concordance rate between the approaches based on CNN and LSTM (0.67-0.68). These approaches show concordance rates a little bit higher (around 2 to 3%) compared with SMV as a baseline approach (0.64-0.66).

Table 4.3: Predictive accuracies for baselines and deep neural network models

Naive Bayes	SVM(Linear)	SVM(RBF Kernel)	CNN	LSTM	Seq2Seq
0.598	0.659	0.664	0.686	0.678	0.718

On the other hand, a model based on Seq2Seq showed the highest concordance rate among all of the models (0.718), higher by 5 to 7% and 3 to 4% compared with SVM and other models, respectively.

Then, results as described above are discussed using Kappa coefficient, which means concordance rate excluding accidental ones. At first, it may be said that LSTM model has achieved sufficiently higher result as the Kappa coefficient for the model shows 0.63. In general, Kappa coefficient of 0.8 or higher is believed to be preferable for utilizing automatic coding discrimination result by a machine in a reliable manner, however, further higher concordance rate is required. In case of Seq2Seq model, on the other hand, Kappa coefficient is 0.723 with great improvement, if not reaching 0.8.

The experiment results above have suggested that Seq2Seq model is superior to other approaches due to consideration for context information. Since Seq2Seq is a model with reply sources entered, it is believed that the improvement in the accuracy has been partly caused by not separate capturing of each contribution but consideration of the context information.

Finally, we analyze for each coding label which misclassification will occur in what case. Table 4.4 shows the precision and recall and F-score of each label for the model using LSTM. “Greeting”, “Agreement” and “Question” can be seen that the F-score is the highest (respectively 0.94, 0.83 and 0.77). These results, since cases can outline deeper easily determined without capturing the meaning of a sentence from the speech is large, it can be considered that also match the human sense. In contrast, “Outside Comments” has the lowest F-score (0.25). This is because statements intended for joke, etc., which have nothing to do with the content to be exchanged correspond to, but in order to judge it, it is thought that it is necessary to deeply grasp the sentence. In addition, “Reply” has lower F-score (0.53). Even using Seq2Seq model, although the F-score of “Reply” is somewhat

improved, it is still found to be low, and by the confusion matrix (Figure 4.2), and it is misclassified into “Agreement” , “Proposal” and “Report” etc. Since “Answer” mostly corresponds to “Question” and F-score of “Question” is high, In the method used this time, we can conclude that extraction of “Reply” is insufficient.

Table 4.4: Precision and recall of each label (LSTM)

	Precision	Recall	F-score
Agreement	0.85	0.81	0.83
Proposal	0.73	0.74	0.73
Question	0.75	0.80	0.77
Report	0.64	0.62	0.63
Greeting	0.94	0.94	0.94
Reply	0.62	0.46	0.53
Outside Comments	0.17	0.47	0.25
Confirmation	0.58	0.74	0.65
Gratitude	0.67	0.67	0.67

4.3 New Coding Scheme

Codes based on speech act which was used in the previous research are factors with difficulty in judgment not only by artificial intelligence but also by manual coding because one code may include another code just like a case that Reply includes a meaning of Agree.

More importantly, in addition to these technical problems, the scheme focuses only on linguistic features that rely on speech act is inadequacies to generically represent a process collaborative learning. In this one-dimensional scheme, it is extremely difficult to answer questions related to the nature of the cooperative process, such as how much each member of the group is involved in problem solving, what kind of division of labor and time management was done, what kind of discussion was being developed, what exchanges of opinions and opinions were shared among members.

From those described above, we propose a new coding scheme so that the automated coding accuracy will improve and that we may understand more accurately and globally collaborative process.

New codes to be proposed are adapted to the current system in reference to a framework in which multi-dimensional codes suggested by Weinberger et al. are used [59]. As shown in Table 4.5, the new coding is composed of 5 dimensions and codes are basically granted by a contribution in chat like the current study. While numerical values including number

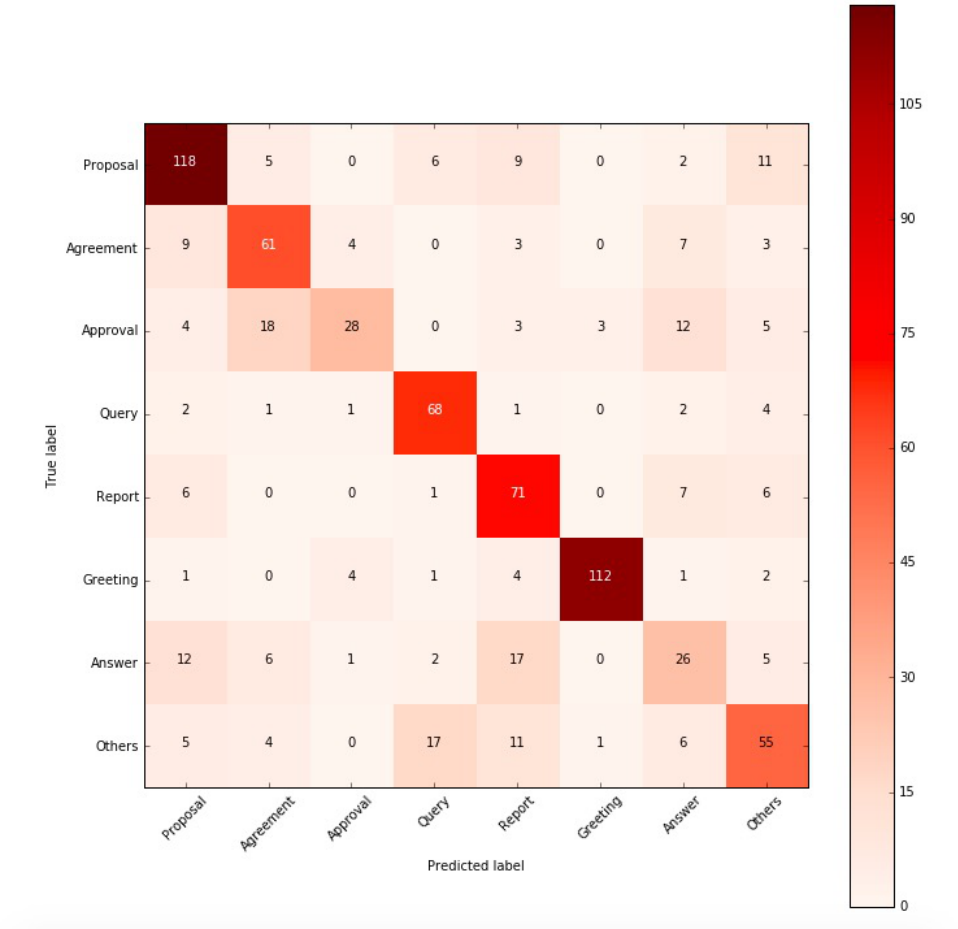

Figure 4.2: Confusion matrix for the Seq2Seq model

of contributions are granted as codes to Participation dimension, one label selected from multiple labels is granted as a code to other four dimensions. Each dimension is described in detail in the following.

Table 4.5: New Coding Scheme

Dimension	Description
Participation	Frequency of participation in argumentation
Epistemic	How to be directly involved in problem solving
Argumentation	Ideal assertion in argumentation
Social	How to cope with others ’ statements
Coordination	How to coordinate to advance discussion smoothly

4.3.1 Participation Dimension

Participation dimension is for measuring degree of participation in arguments. As this dimension is defined as quantitative data including mainly number of contributions and its letters, time of contributions, and interval of contributions, coding is performed by statistical processing on the database while requiring neither manual nor artificial intelligent coding. The list is shown in Table 4.6.

Since Participation dimension labels handle number of specific contributions, it is possible to analyze quantitatively different aspects of participation in conversations but impossible to perform qualitative analysis such as whether the conversation contributed to problem solving.

Table 4.6: Participation Dimension

Category	Description	Example
Number of contributions	Number of contributions of each member during sessions	59 times
Number of letters of a contribution	Number of letters during a single speech	15 letters
Time for contribution	Time used for a contribution	2017/8/21 15:15:01
Interval of contributions	Time elapsed since last contribution	3:01.05

4.3.2 Epistemic Dimension

Representing whether each contribution is directly associated with resolution of problems as a task, it is classified depending on contents of the contributions as shown in Table 4.7. This dimension ' s codes are granted to all contributions.

Weinberger and Fischer ' s scheme has 6 categories to code epistemic activities, which consist in applying the theoretical concepts to case information. But, as shown in Table 4.7, we set only two categories here, because we want to give generality by which we can handle as many problem solving types as possible. "On Task" here refers to contributions directly related to resolution of assigned tasks and such contributions with contents as shown below belong to "Off Task".

- Contributions to ask meaning of problems and how to proceed with them

- Contributions to allocate different tasks to members
- Contributions regarding the system

Since Epistemic dimension represents whether directly related to problem solving, it works as the most basic code for qualitative analysis. In case of less “On Task ” labels, for example, it is believed that almost no effort has been made for the task.

Besides, labels of Argument and Social dimensions are assigned when Epistemic dimension is “On Task ” , whereas those of Coordination dimension are assigned only when it is “Off Task ” .

Table 4.7: Labels in Epistemic Dimension

Label	Description
On Task	Contributions directly related to problem solving
Off Task	Contributions without any relationship with problem solving
No Sense	Contributions with nonsensical contents

4.3.3 Coordination Dimension

Coordination dimension code is assigned only when Epistemic code is “Off Task ” and it is also assigned to such contributions that relate to problem solving not directly but indirectly. A list of Coordination dimension labels is shown in Table 4.8 but the labels are assigned not to all contributions of “Off Task ” but just one label is assigned to such contributions that correspond to these labels. In addition, in case of replies to contributions with Coordination dimension labels assigned, labels of the same Coordination dimension are assigned.

Here, “Task Division” refers to a contribution to decide who to work on which task requiring division of tasks for advancing problem solving. “Time Management” is a contribution to coordinate degree of progress in problem solving, and for example, such contributions fall under the definition that “let’s check it until 13 o’clock,” and “how has it been in progress?” “Meta contribution” refers to a contribution for clarifying what the problem is when intention and meaning of the problem is not understood. “Technical Coordination” refers to questions and opinions about how to use the CSCL System. “Proceedings ” refer to contributions for coordinating the progress of the discussion.

Since Coordination dimension code is granted to such contributions that intend to resolve problems smoothly, it is believed to be possible to predict progress in arguments by analyzing timing when the code was granted. Further, in case of less codes of Coordination

dimension, it may be predicted that smooth relationship has not been created within the group.

On the other hand, if a large number of these codes were granted in many groups, it may be understood that there exists any defect in contents of the task or system.

Table 4.8: Labels of Coordination Dimension

Label	Description
Task Division	Splitting work among members
Time Management	Check of temporal and degree of progress
Technical Coordination	How to use the system, etc.
Proceedings	Coordinating the progress of the discussion.

4.3.4 Labels of Argument Dimension

Labels of Argument dimension are provided to all contributions, indicating attributes such as whether each contribution includes the speaker ' s opinion and whether the opinion is based on any ground. Labels of this dimension are provided to just one contribution content without considering whether any ground was described in other contribution.

Table 4.9 shows a list of Argumentation dimension codes. Presence/absence of ground mentioned here depends on whether any ground to support an opinion is presented requiring any credibility of the ground presented. In addition, limiting condition represents whether a suggested opinion is asserted to be applicable to all of situations handled as a task or just part of them. For example, it is applicable to such cases in which any paragraph such as "in case of" or "compared with" is included. "Non-Argumentative Moves" refers to contributions without any opinion and therefore, simple questions are included in this tag. Also, as a logical consequence, this label is assigned to all off-task contribution in the Epistemic dimension.

Argumentation dimension code is capable of analyzing significance of contribution contents. Therefore, an argument with just "Simple Claims" may be understood as a superficial one. In comparison with Weinberger and Fischer ' s scheme, we do not set for now the categories of macro-level dimension in which single arguments are arranged in a line of argumentation such as arguments, counterarguments, reply, for the reason that it seems difficult that the automatic coding by deep learning methods for this macro dimension works correctly.

Table 4.9: Labels of Argument Dimension

Label	Description
Simple Claim	Simple opinion without any ground
Qualified Claim	Opinion based on a limiting condition without any ground
Grounded Claim	Opinion based on grounds
Grounded and Qualified claim	Opinion with limitation based on grounds
Non-Argumentative Moves	Contribution without containing opinion (including questions)

4.3.5 Labels of Social Dimension

Labels in Social dimension are provided when Epistemic code is "On Task" but they are provided not to all contributions "On Task" but to a contribution which conforms to Epistemic code. This dimension represents how each contribution is related to those of other members within the group. Therefore, it is required to understand not only a contribution but also the previous context. Table 4.10 shows a list of labels of the dimension.

In this case, "Externalization" refers to contributions without reference to other ' s contributions and it is granted to contributions to be an origin of arguments mainly at the start of argument on a topic. "Elicitation" is granted to such contributions that request others for extracting information including question. "Consensus Building" refers to contributions that express certain opinion in response to other ' s contribution and they are classified into the three codes below. "Quick Consensus Building" is granted to such contributions that aim to form prompt consensus with other ' s opinion. It is granted to a case to give consent without any specific opinion. "Integration-Oriented Consensus Building" is granted to such contributions that intend to form consensus with other ' s opinion while adding one ' s own opinion. "Conflict-Oriented Consensus Building" is granted to such contributions that confront with other ' s opinion or request revision of the opinion.

Social dimension contains a sub-dimension called "Refer" to represent which contribution is referenced and number of referenced contribution is usually granted as a code. Codes of "Refer" dimension is granted only if Social dimension code is "Consensus Building". Since Social dimension code represents involvement with others, it may be understood how actively the argument was developed or whose opinion within the group was respected by analyzing Social dimension codes. For example, it may be assumed that arguments with frequent "Quick Consensus Building" result in accepting all opinions provided with almost no deep discussion.

Table 4.10: Labels of Social Dimension

Label	Description
Externalization	No reference to other ' s opinion
Elicitation	Questioning the learning partner or provoking a reaction from the learning partner
Quick Consensus Building	Prompt consensus formation
Integration-Oriented Consensus Building	Consensus formation in an integrated manner
Conflict-Oriented Consensus Building	Consensus forming based on a confrontational stance
Summary	Statement listing or quoting contributions

4.3.6 Relationships among the Dimensions

While Participation dimension code is automatically generated by a system based on logs of contributions in the new coding scheme, manual granting by a coder is required for other codes. In addition, which of codes of Argumentation, Social or Coordination dimension to be granted is determined according to a result of Epistemic dimension codes.

Therefore, the coder grants Epistemic dimension code by analyzing contribution contents and Participation dimension codes. Subsequently, in a case that Epistemic dimension code is "On Task", Argumentation and Social dimension codes are granted. In addition, in a case that Social dimension code is "Consensus Building", a contribution number is granted as "Refer" since it is based on a reference source contribution without exception. In a case that Epistemic dimension code is "Off Task", Coordination dimension code is granted. The relationships among the dimensions is shown in Table 4.3

4.4 Experiments and Results

We trained a new dimension using the Seq2Seq-based architecture, which is the most accurate among the methods described in Section 4.2. Separate data was prepared for each dimension, independent learning was performed four times in total, and four separate learned models were created. The sizes of the data are 8,460 for the Epistemic dimension, 7,795 for the Augment dimension, 3,510 for the Coordination dimension, 2,619 for the Social dimension. These data was used for learning the model.

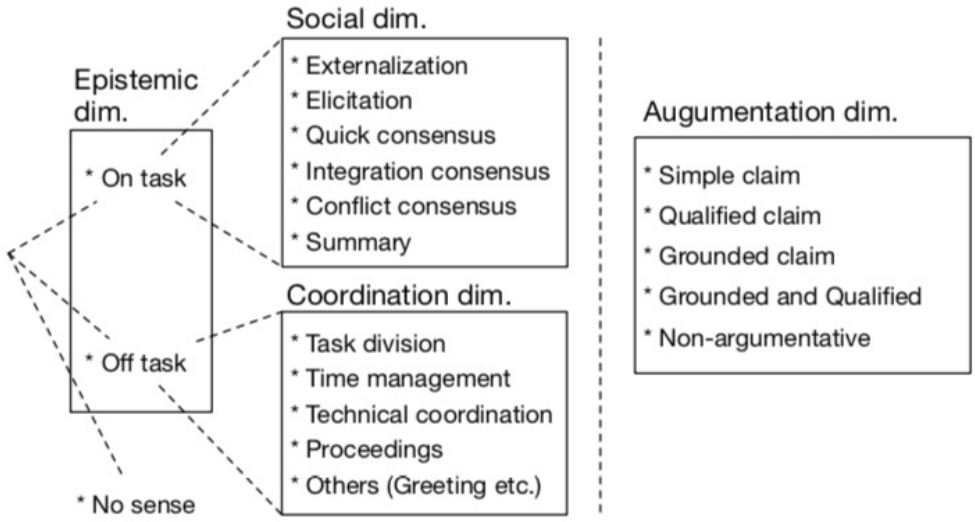

Figure 4.3: Relationships among the Dimensions

4.4.1 Manual Coding Results

First of all, as described above, coding is performed manually for each contribution in order to to make models using deep learning. For all contributions, two coders assigned labels to each, and data in which two labels coincided were used as correct true labels. The ratios of each label in each dimension among all contributions are shown in the following. From the viewpoint of machine learning, it can be said that these graphs show the ratios of labels that occupy in all true labels.

The ratios of “On Task” and “Off Task” in the Epistemic dimension are shown in Figure 4.4. In our dataset, the “On Task” contributions were a bit fewer than the “Off Task”. Generally, it is a typical task of binary classification and is thought to be relatively easy to predict using machine learning. This implies that, at least from the view point of the conversation log, the cost of the communication was more than the cost of discussion in group work. Although this result is just an instance obtained by applying our CLCS system to the actual group works for limited lectures, we can at least conclude that the communication cost is not small in a group work.

The labels in the Argument dimension are assigned independently of other dimensions. Thus, its domain spans both the “On Task” and the “Off Task” contributions. As shown in Figure 4.5, the label ”Non-Argumentative Moves” occupied more than 60% of all. The label “Simple Claim” occupied the second percentage. “Non-Argumentative Moves” and “Simple Claim” account for more than 95% in total Therefore, from the viewpoint of machine learning, generally, it is considered relatively easy to classify for the above two, but

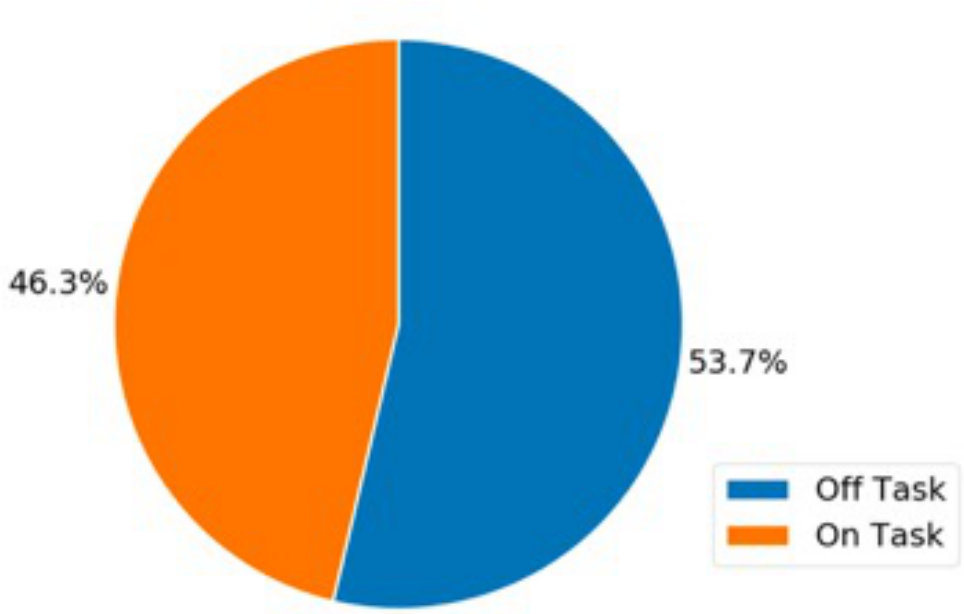

Figure 4.4: Ratio in the Epistemic dimension

for the remaining “Claim”, it can be considered that learning can not be sufficiently done due to the data number problem. To assess the discussion of the group work, at least it is necessary to remove the “Non- Argumentative Moves” contributions and pay attention to which kind of claim is presented, even if almost every claim can be classified into the “Simple Claim”. Therefore, the automatic coding for this dimension is as valuable as for the other three dimensions.

With respect to the Coordination dimension, the domain of which is the “Off-Task” contributions, the most of them are assigned to “Other” as Figure 4.6 shows. The contributions labeled “Other” consist of short sentences that are not significant for neither discussion nor coordination of the group work. The representative examples are greetings and kidding. Meanwhile, the statistics show that the contributions except for “Other” also occupies more than a quarter. Since these kinds of contributions are related to coordinating tasks in the group work, they can be thought as important contributions for the assessment.

Figure 4.7 shows the ratios of the labels in the Social dimension. The label “Externalization” accounted half of the “On Task” contributions. The “Quick Consensus Building” followed it. Meanwhile, the ratios of the “Summary” and the “ Consensus Buildings” except for the “Quick” one were small. These statistics show that the actual discussion mainly consisted of expressions of their opinions. Although we found that the contributions build-

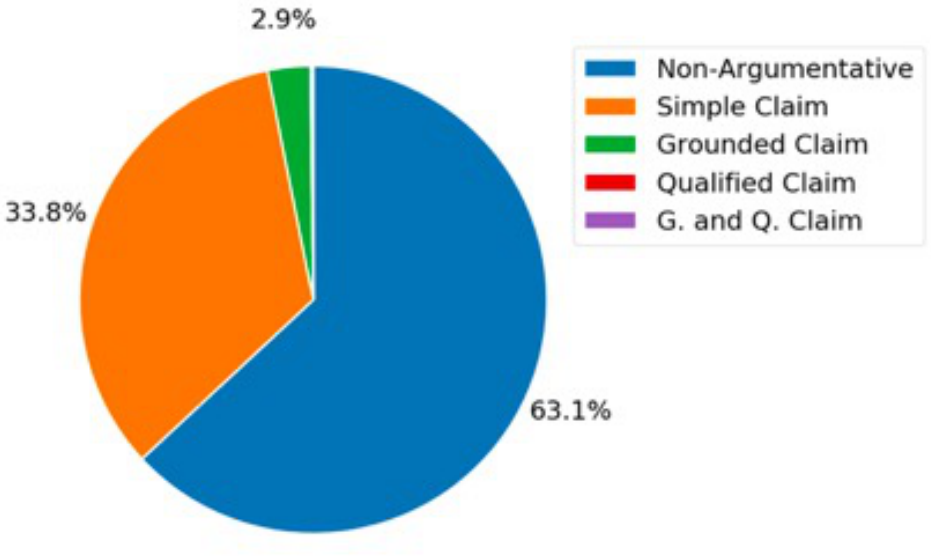

Figure 4.5: Ratio in the Argument dimension

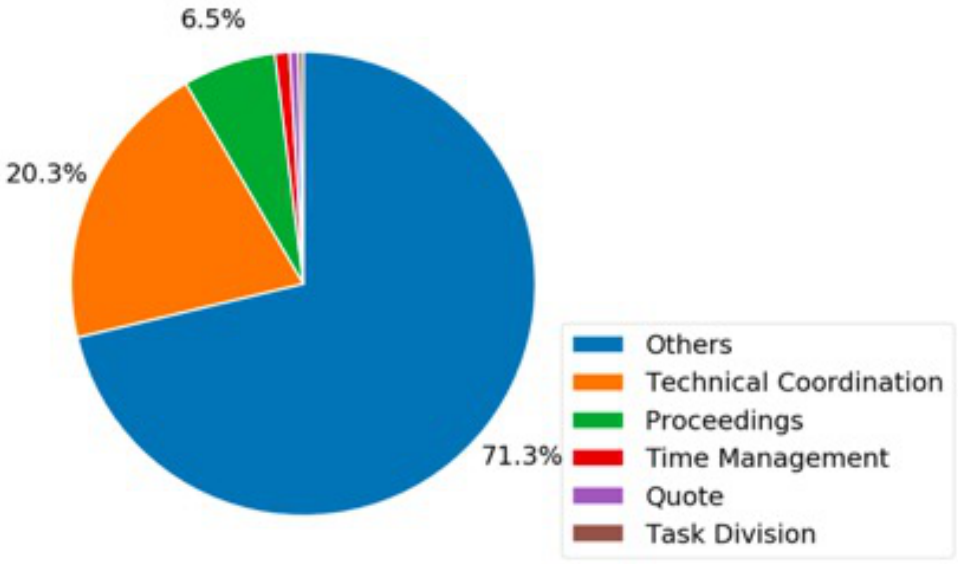

Figure 4.6: Ratio in the Coodination dimension

ing consensus rarely come up in a real group work, we believe that they are the important keys for the discussion. Thus, we may can weight them when we assess the contribution to the discussion by students.

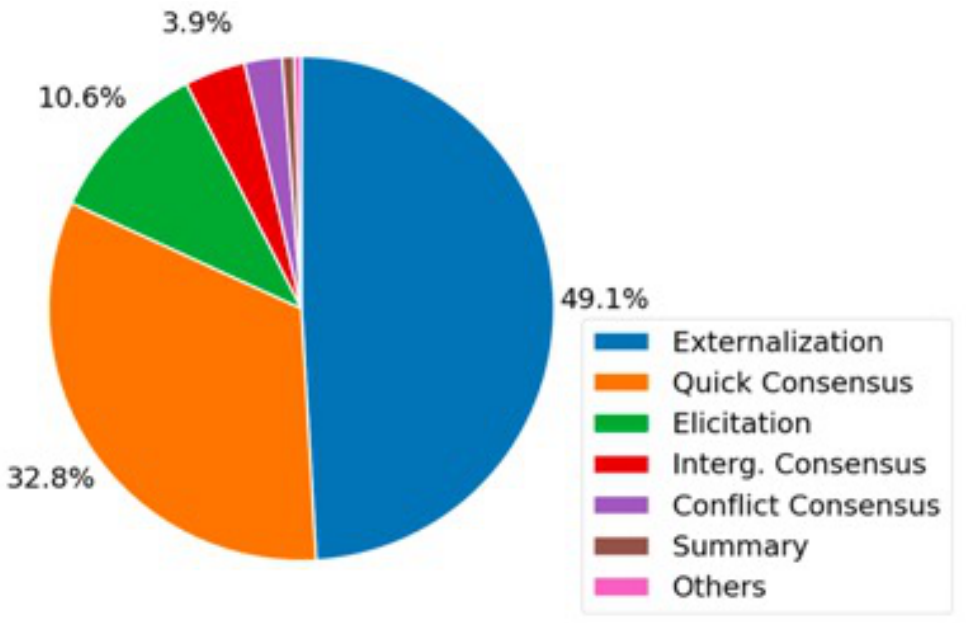

Figure 4.7: Ratio in the Social dimension

4.4.2 Results for each dimensions

We applied the learned DNN model to the test data and let us predict the label of each dimension.

Table 4.11: Precision and Recall for the Epistemic dimension

	Precision	Recall	F-score	Support
On Task	0.90	0.91	0.90	390
Off Task	0.92	0.91	0.91	456
Average(Micro) / Total	0.91	0.91	0.91	846

The results of the experiments show that the “On Task” and “Off Task” can be classified correctly with sufficiently high accuracy (Figure 4.8). The Seq2Seq based model achieves more than 90% in both precision and recall (Table 4.11). Since the coincidence ratio by two human coders is 91%, we can say that the accuracy of automatic coding, which is comparable to human beings was obtained for the “Epistemic” dimension.

The classification accuracy is also high for the Argument dimension. The micro-averaged F-score is 87% (Table 4.12). Especially, the F-score for the label “Non-Argumentative Moves” is high sufficiently (92%), which means that our model can surely recognize whether the contribution has any substantial meaning as a claim or not. On the other

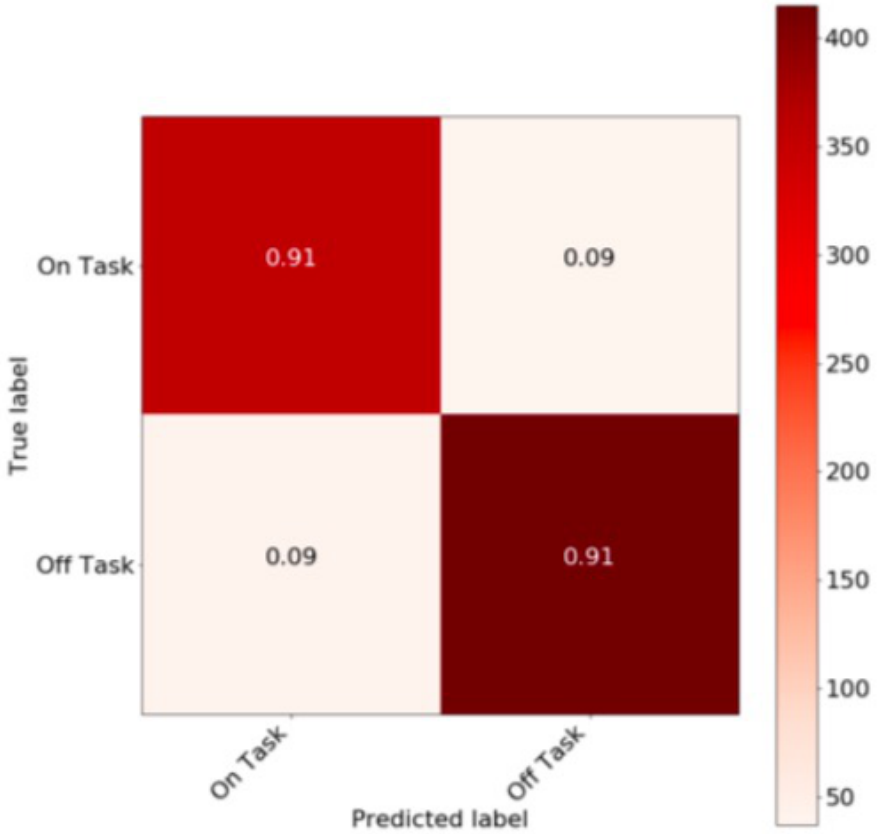

Figure 4.8: Confusion matrix for the Epistemic dimension

Table 4.12: Precision and Recall for the Argumentation dimension

	Precision	Recall	F-score	Support
Non-Argumentative	0.87	0.97	0.92	491
Simple Claim	0.89	0.72	0.80	264
Grounded Claim	0.58	0.52	0.55	21
Qualified Claim	0.00	0.00	0.00	1
Average(Micro) / Total	0.87	0.87	0.87	777

hand, while the precision for the "Simple Claim" is high (89%), the recall for it is low (72%). According to the confusion matrix shown in Figure 4.9, a quarter of the "Simple Claim" is misclassified into the "Non-Argumentative Moves". This is because it is difficult to distinguish contributions that have a very small opinion from that have no opinions.

Regarding the Coordination dimension, our model also achieved high classification accuracy. Seeing that the number of supports varies greatly among the labels, we should

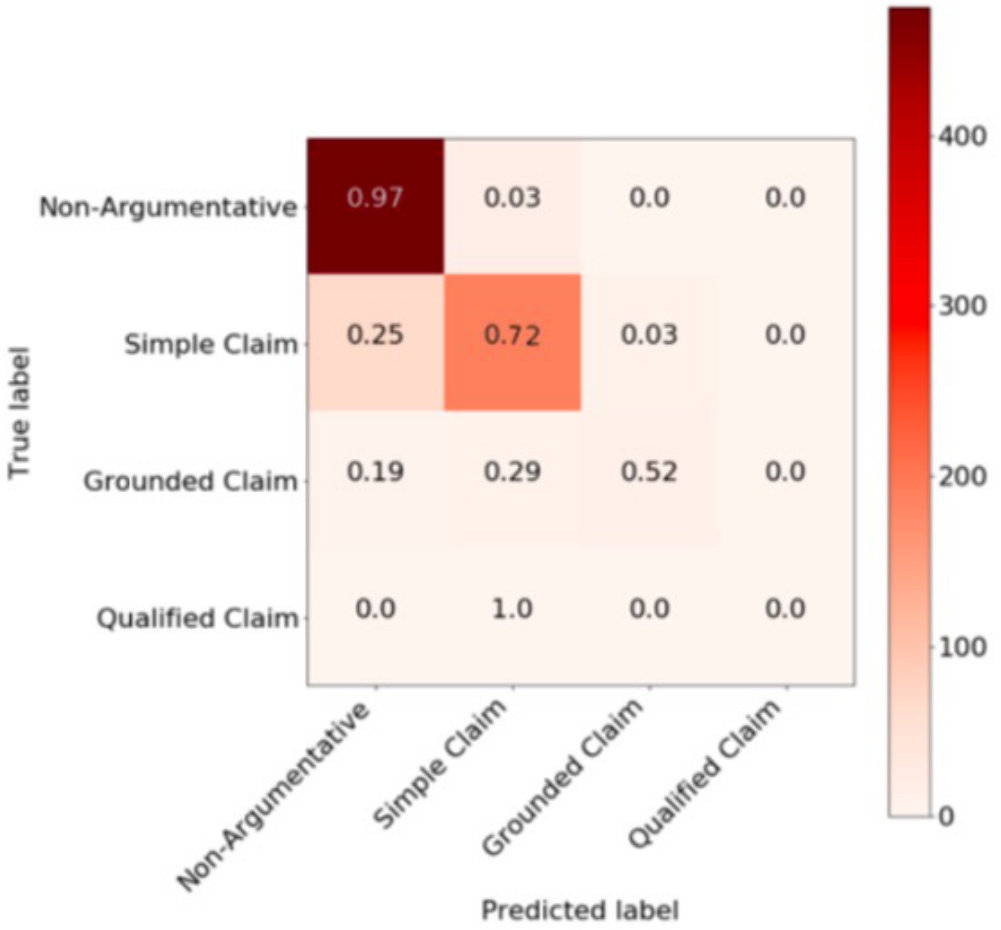

Figure 4.9: Confusion matrix for the Argumentation dimension

Table 4.13: Precision and Recall for the Coordination dimension

	Precision	Recall	F-score	Support
Others	0.91	0.91	0.91	242
Technical Coordination	0.81	0.80	0.81	82
Proceedings	0.58	0.70	0.64	20
Time Management	0.33	0.25	0.29	4
Quote	0.00	0.00	0.00	1
Task Division	0.00	0.00	0.00	2
Average(Micro) / Total	0.85	0.86	0.85	351

evaluate the classification ability of the model by the micro-averaged accuracies over all coding labels. As Table 4.13 shows, the micro-averaged F-score was 85%. According to the results for each label (Figure 4.10), the following is observed. The major labels such as

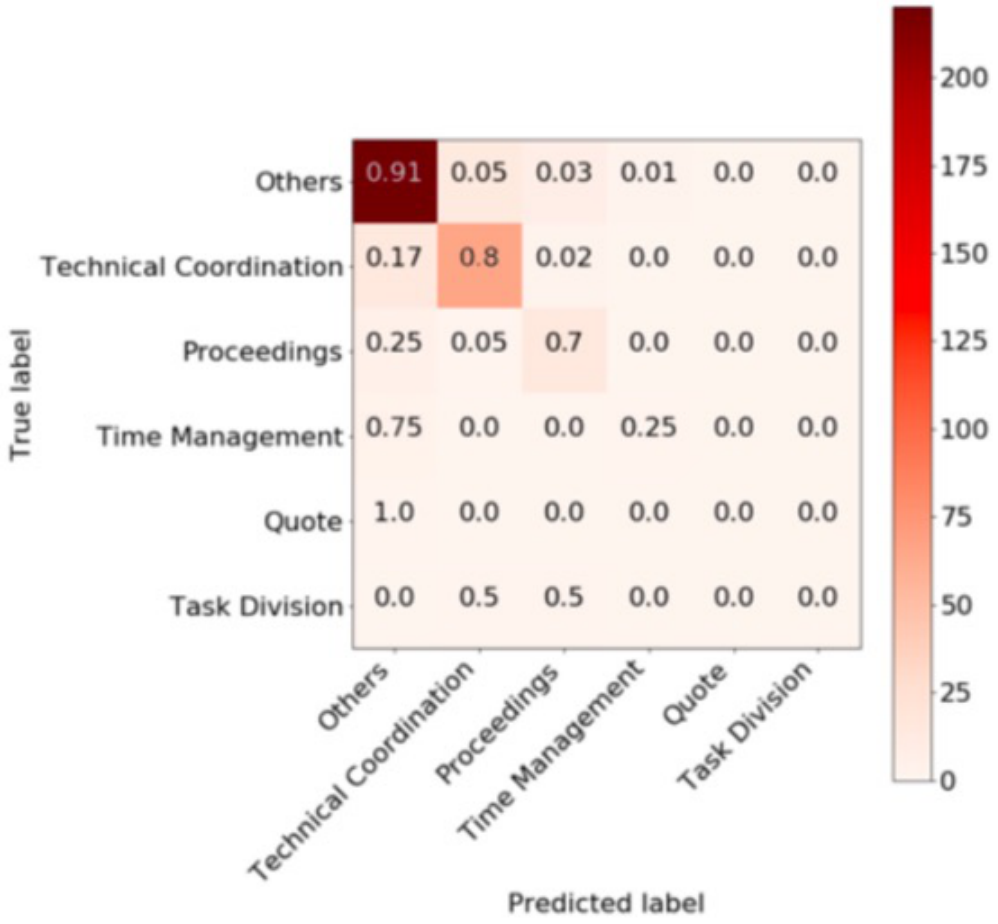

Figure 4.10: Confusion matrix for the Coordination dimension

"Other" and "Technical Coordination" are classified correctly with high precisions, while the minor labels such as "Time Management", "Quote" and "Task Division" are not. Because the data for those miner labels are very limited, which have less than 50 contributions, it is quite difficult to learn them accurately. One of our future issues is to find some way to deal with those sparse labels.

Comparing to the other dimensions, the accuracy was relatively low for the Social dimension. The F-score was 70% (Table 4.14). Since labeling the Social dimension sometimes needs understanding the deep meaning of the contribution and the background story of the discussion, it seems to be difficult for machines to learn them correctly with limited data. According to Figure 4.11, the recall of the label "Externalization " is especially low (61%), while those of "Quick Consensus " and "Elicitation" are high sufficiently (93% and 97%, respectively). According to the confusion matrix in Figure 4.11, there is a major reason that worsen the accuracy; the "Externalization" labels are easily misclassified to the "Quick Consensus" and to the "Elicitation", but not vice versa. This fact also explains the

Table 4.14: Precision and Recall for the Social dimension

	Precision	Recall	F-score	Support
Externalization	0.86	0.61	0.72	127
Quick	0.71	0.93	0.81	88
Elicitation	0.56	0.97	0.71	29
Interg. Consensus	0.17	0.14	0.15	7
Conflict Consensus	0.00	0.00	0.00	6
Summary	0.00	0.00	0.00	3
Others	0.00	0.00	0.00	2
Average(Micro) / Total	0.75	0.72	0.70	262

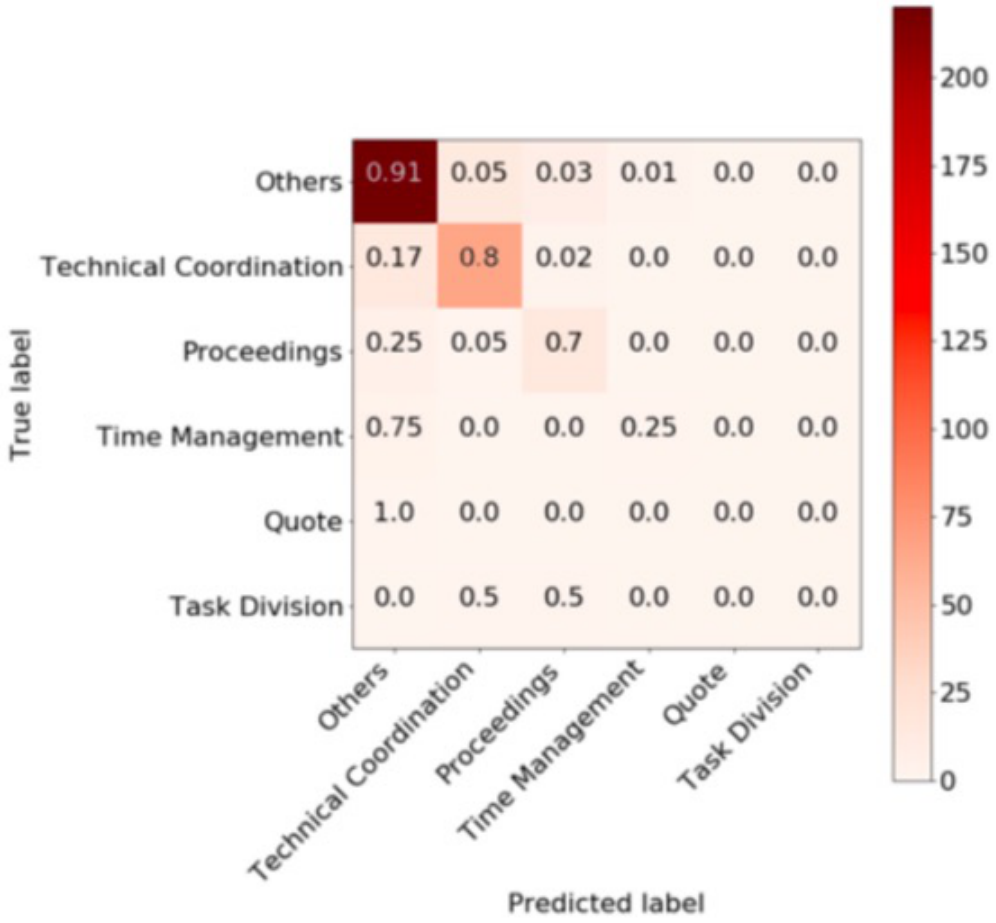

Figure 4.11: Confusion matrix for the Social dimension

reason why the precisions for the “Quick Consensus” and the “Elicitation” are low though the recalls for them are high. To improve the result, it is necessary to pursue the causes of these two types.

4.5 Verification of the proposed method

For the proposed new coding scheme, we will automatically code the actual chat data and consider what kind of analysis is possible.

4.5.1 Conversation Dataset

Table 4.15, shows the details of chat data to be automatically coded in this verification. The final theme of the lecture is to submit for each group, and it is called “propose a new educational TV program”, but it includes “Title”, “Learning Task”, “Subject”, “Program Contents”, “Ingenuity Points” and “Features”.

In addition, each group’s submissions are evaluated by each faculty in three levels (Excellent, Average and Poor) with “Concreteness”, “Ingenious” and “Appropriateness”, and the total is evaluated as “Overall”. “Concreteness” means whether the program contents can be imagined with feasibility from the proposal content, “Ingenious” is whether unique in approach and concept, “Appropriateness” was evaluated to what extent the relevancy between the program content and the program object was compatible. Table 4.16 shows the number of groups with each evaluation.

Table 4.15: Contributions data used in this study

Date and Time	July 17 and 24, 2017
Lecture Name	Educational Media Theory
Task Contents	Proposal of educational TV program
Learning Time	Total 2 hours
Number of Students	138 students
Member of Groups	3 students
Number of Groups	46 groups
Number of Contributions	2743 contributions

Table 4.16: Number of Evaluation of each groups

	Excellent	Average	Poor
Overall	7	20	19
Concreteness	10	18	18
Ingenious	13	19	14
Appropriateness	12	25	9

4.5.2 Automatic coding result

Using the four learned models obtained in this experiment, automatic coding processing was performed for all 2743 contributions. Tables 4.17 to 4.20 show the number of contributions of each label of four dimensions.

Table 4.17: Automatic coding results of Epistemic dimension

Label	Number of Contributions
On Task	1633
Off Task	1110

Table 4.18: Automatic coding results of Argumentation dimension

Label	Number of Contributions
Simple Claim	1082
Non-argumentative moves	1638
Grounded Claim	23
Grounded and Qualified claim	0
Qualified Claim	0

Table 4.19: Automatic coding results of Coordination dimension

Label	Number of Contributions
Others	2368
Technical coordination	360
Proceedings	15
Time management	0
Task division	0

4.5.3 Evaluation of Submission and Contributions

Tables 4.21 to 4.24 show the average number of labels given for each dimension evaluation. Also, Table 4.25 shows correlation coefficients with the number of contributions of each label, with each evaluation of “Overall”, “Concreteness”, “Ingenious” and “Appropriateness” being Excellent = 3, Average = 2 and Poor = 1. Bold items are items that

Table 4.20: Automatic coding results of Social dimension

Label	Number of Contributions
Externalization	2170
Elicitation	152
Quick consensus building	421
Integration-oriented consensus building	0
Conflict-oriented consensus building	0
Summary	0
Others	0

Table 4.21: Evaluation of Submission and Average number of Contributions (Epistemic)

(a) Overall			
Evaluation	On Task	Off Task	Total
Excellent	38.7	22.7	61.4
Average	35.4	23.2	58.6
Poor	33.7	24.7	58.4
(b) Concreteness			
Evaluation	On Task	Off Task	Total
Excellent	40.6	22.6	63.2
Average	33.5	23.7	57.2
Poor	33.9	24.4	58.3
(c) Ingenious			
Evaluation	On Task	Off Task	Total
Excellent	39.6	23.8	63.5
Average	35.9	25.3	61.2
Poor	30.1	21.5	51.6
(d) Appropriateness			
Evaluation	On Task	Off Task	Total
Excellent	36.9	21.2	58.2
Average	33.2	24.9	58.2
Poor	38.3	23.8	62.1

the absolute value of 0.2 or more weak correlation of the correlation coefficients. From this result, we can see that “On Task” of Epistemic dimension , “Non-argumentative” of Argumentation dimension “Others” and “Technical Coordination” of Coordination dimension and “Externalization” of the Social dimension, there is a positive correlation with the number of contributions, and the more the five labels are, the higher the evaluation of the

Table 4.22: Evaluation of Submission and Average number of Contributions (Argumentation)

(a) Overall				
Evaluation	Non-argumentative	Simple Claim	Grounded Claim	Total
Excellent	33.7	27.6	0.1	61.4
Average	35	22.9	0.6	58.6
Poor	35.8	22.2	0.4	58.4
(b) Concreteness				
Evaluation	Non-argumentative	Simple Claim	Grounded Claim	Total
Excellent	34.4	28.3	0.5	63.2
Average	34.9	21.8	0.5	57.2
Poor	35.8	22.1	0.4	58.3
(c) Ingenious				
Evaluation	Non-argumentative	Simple Claim	Grounded Claim	Total
Excellent	38.1	24.9	0.5	63.5
Average	35.9	24.7	0.6	61.2
Poor	31.4	19.9	0.4	51.6
(d) Appropriateness				
Evaluation	Non-argumentative	Simple Claim	Grounded Claim	Total
Excellent	31.4	26.4	0.3	58.2
Average	36.5	21.1	0.6	58.2
Poor	36.3	25.4	0.3	62.1

“Ingenious” is. About “Ingenious”, we believe that how much the conversation took place within the group is important. About “Overall”, “Concreteness” and “Appropriateness”, there is a negative correlation with “Elicitation” of Social dimension , and as the number of contributions, the evaluation is lower.

On the other hand, in order to compare whether the difference in the number of contributions of each member within the group is related to the evaluation of the submission, the variation coefficient of the number of contributions for each label of each member in the group was obtained. When the coefficient of variation is high, the number of conversations in the group is large, such as speaking remarkably by only one person on the label. Table 4.26 shows the correlation coefficient between the variation coefficient of each label and the evaluation of each item. Bold items are items that the absolute value of a weak correlation 0.2 or more correlation coefficients. All items with high correlation coefficients

Table 4.23: Evaluation of Submission and Average number of Contributions (Coordination)

(a) Overall				
Evaluation	Others	Technical Coordination	Proceedings	Total
Excellent	53.7	7.4	0.3	61.4
Average	50.4	7.8	0.3	58.6
Poor	50.5	7.5	0.4	58.4
(b) Concreteness				
Evaluation	Others	Technical Coordination	Proceedings	Total
Excellent	54.1	8.8	0.3	63.2
Average	49.9	6.9	0.3	57.2
Poor	50.3	7.7	0.3	58.3
(c) Ingenious				
Evaluation	Others	Technical Coordination	Proceedings	Total
Excellent	54.3	8.8	0.3	63.5
Average	52.7	8.2	0.3	61.2
Poor	45.5	5.7	0.4	51.6
(d) Appropriateness				
Evaluation	Others	Technical Coordination	Proceedings	Total
Excellent	50.2	7.8	0.2	58.2
Average	49.8	7.9	0.5	58.2
Poor	55.2	6.8	0.1	62.1

are negative correlations, and when the difference in the number of conversations in the group is large, the evaluation gets worse. But, there is a correlation between the deviation of the number of contributions of "Quick Consensus" and the evaluation, indicating that if the number of contributions of "Quick Consensus" is biased, the evaluation tends to be worse.

Table 4.27 excerpts the contributions which are actually given the "Elicitation" and "Quick Consensus" of Social dimension. "Elicitation" is a remark to ask other student's remarks by question, and according to the content of the remarks, it is considered that the students will be confused about the contents of the task and how to proceed. "Quick Consensus" is a remark to aim for an immediate agreement on the opinions of other student, etc. According to the content of the remarks, it is predicted that the conversation in the group will not get excited.

Table 4.24: Evaluation of Submission and Average number of Contributions (Social)

(a) Overall				
Evaluation	Quick Consensus	Externalization	Elicitation	Total
Excellent	8	51.4	2	61.4
Average	9.4	46.4	2.8	58.6
Poor	9.3	44.9	4.3	58.4
(b) Concreteness				
Evaluation	Quick Consensus	Externalization	Elicitation	Total
Excellent	8.4	52.4	2.4	63.2
Average	9.2	45.1	2.9	57.2
Poor	9.4	44.7	4.2	58.3
(c) Ingenious				
Evaluation	Quick Consensus	Externalization	Elicitation	Total
Excellent	8.8	51.8	2.8	63.5
Average	9.2	48.8	3.2	61.2
Poor	9.3	38.6	3.8	51.6
(d) Appropriateness				
Evaluation	Quick Consensus	Externalization	Elicitation	Total
Excellent	7.5	48.7	2	58.2
Average	10.1	44.3	3.8	58.2
Poor	8.6	49.9	3.7	62.1

4.5.4 Discussion

The proposed method, the multi-dimensional automatic coding also for the new chat data that is capable revealed. In addition, based on the evaluation of each group by the teacher, it was possible to find problem points in the group earlier and to search for solutions.

4.6 Conclusion

In this study, we proposed a newly designed coding scheme with which we tried to automate time-consuming coding task by using deep learning technology. We have constructed a new coding scheme with five dimensions to analyze different aspects of the collaboration process. After manually coding a large volume dataset, we proceeded to the machine learning of this dataset using Seq2seq model. Then, we evaluated the accuracy of this au-

Table 4.25: Correlation coefficient between the submission evaluation and the number of contributions

Label	Overall	Concreteness	Ingenious	Appropriateness
Epistemic Dimension				
On Task	0.11	0.15	**0.24**	-0.02
Off Task	-0.09	-0.08	0.11	-0.11
Argumentation Dimension				
Non-argumentative	-0.05	-0.04	**0.20**	-0.14
Simple Claim	0.14	0.18	0.17	0.05
Grounded Claim	-0.05	0.03	0.05	-0.02
Coordination Dimension				
Others	0.05	0.07	**0.20**	-0.09
Technical Coordination	0	0.06	**0.26**	0.06
Proceedings	-0.06	-0.02	-0.09	0
Social Dimension				
Quick Consensus	-0.08	-0.1	-0.04	-0.12
Externalization	0.11	0.14	**0.27**	-0.01
Elicitation	**-0.37**	**-0.31**	-0.16	**-0.26**
Total	0.04	0.07	**0.22**	-0.06

tomatic coding in each dimension. Except some typical types of the misclassifications, the results were overall very good. These results indicate with certainty that we can introduce this model to authentic educational settings and that even for large classes that have many students, we can perform real-time monitoring of learning process or ex-post analysis of big educational data.

As for the future research directions, we may have two approaches to pursue. The first approach is about some typical misclassifications in the "Social" Dimension. To improve prediction accuracy, one could make more explicit and comprehensible the referential relation between a contribution and others even for the machines, if one indicates contributions to which a contribution refers. For example, with regard to the typical misclassification mentioned above between "Externalization" and "Quick Consensus" or "Elicitation", since contributions labeled "Externalization" have no reference to other contributions, we can hope to effectively reduce these misclassifications with this kind of indicator. In addition, as the next step of this thesis, it seems to be worth trying to compare the accuracy using DNN models other than Seq2seq and other network structures such as memory networks [60]. The second approach concerns the intrinsic structure of our coding scheme. Since the scheme contains different dimensions and under each dimension different labels

Table 4.26: Correlation coefficient between the submission evaluation and the deviation of the number of contributions

Label	Overall	Concreteness	Ingenious	Appropriateness
Epistemic Dimension				
On Task	-0.06	-0.07	-0.07	-0.01
Off Task	0.05	0.05	-0.12	0.09
Argumentation Dimension				
Non-argumentative	0	0.07	-0.10	0.10
Simple Claim	0.14	0.09	0.06	0.14
Grounded Claim	0.01	0.02	0.10	0.03
Coordination Dimension				
Others	0.18	0.12	0.06	0.18
Technical Coordination	0.10	0.07	-0.04	0.03
Proceedings	-0.05	0.01	-0.05	0.03
Social Dimension				
Quick Consensus	**-0.31**	**-0.28**	**-0.28**	**-0.26**
Externalization	0.18	0.13	0.19	0.10
Elicitation	0.05	0.01	-0.07	0.17
Total	0.14	0.08	0.05	0.14

Table 4.27: Contributions of “Elicitation” and “Quick Consensus”

”Elicitation” example 1	結局どれですか？
”Elicitation” example 2	内容はどうしますか？
”Elicitation” example 3	提出しますか？
”Elicitation” example 4	どうしますか？
”Quick Consensus” example 1	それはいいですね
”Quick Consensus” example 2	そうですね
”Quick Consensus” example 3	タイトル難しいですね。。。
”Quick Consensus” example 4	じゃあこれで決定ですね。

are hierarchically organized, it is very interesting to discover not only correlations among dimensions, but also among labels belonging to different dimensions [61]. If we can input the information about the correlation between such labels in some form at the time of automatic classification, the accuracy of automatic coding can be further improved.

Chapter 5

Conclusion

In this chapter, we summarized the research results and looking forward to the future research works.

5.1 Research Results

The existing research results show that most of the feature sets used in semantic relation classification are complex. Although the better classification accuracy could be obtained, but the calculation costs are usually very higher. To solve this problem, We purposed to find a feature for semantic relation classification that takes both lightweight and high accuracy into account.

Our study first starts with automatically extracting semantic relationships. We want to find a method to extract the unknown semantic relation instance from Wikipedia data by using the known semantic relation instance. In many cases, we found that many information related to semantic relations are hide in the sentences. If we can use these information separately, we could solve our problems. Next, we analyzed various features commonly used in existing studies in semantic relation classification, and raised a new feature independent on the external resources According to the experimental results, we obtained the following results.

(1) A method for automatic extraction of semantic relations

A new method is proposed to automatically extract "part/material concept" using Associative Concept Dictionaries and Wikipedia data. For the semantic relation of parts/material, we set up a group of teaching data to train the SVM classifier by associated the known "part/material" examples in the Associative Concept Dictionaries and combined with the text data of Wikipedia articles. Then we obtained new "part/material concept" by using the

trained classifiers to classify the unknown instances of "part/material concept", Finally, we verified our method with 869 pieces of validation data, and got the classification result with 80% accuracy.

(2) A feature applicable to semantic relation classification that is lightweight and highly accurate

We drew inspirations from the study of distributed representation of words (word vectors) and proposed a new distributed representation to be called *substring vector*. This is a distributed representation of a sequence of words between two nouns in a sentence. According to the neural network language model, we get the distributed representation of words. There are a lot of semantic relation information hiding in this distributed representation. Base on a sentence with a sequence of words between two nouns, we combined the distributed representation of the words in a word sequence and build a *substring vector*. And because this distributed representation does not depend on any external data sources of semantic relations, the dimensions of the feature vector are also relatively low. Through the verification experiments, the proposed feature in Task8 for semantic relation classification, The classification results with accuracy of 78.10% were obtained, which was a better result compared with similar studies.

(3) A weighting method for feature vectors based on word frequency

For the feature vectors used in semantic relation classification, we propose a weighting method based on word frequency. We supposed that if a word often appears in a sequence of words between two nouns, it then carries more potential semantic relation information that can be used for semantic classification. In addition, we propose three weighting methods for comparison. Experiments show that in the case of less training data, our proposed weighting method is far superior to the other three. Moreover, the accuracy of the semantic classification results of previous Task8 could be improved by 1% to 3%.

Based on the above research results, we construct feature vectors for semantic relation classification by using the newly proposed *substring vector*. Through this simple distributed representation, we successfully extract the underlying semantic relationship information in sentences. Compared to the existing methods that use a lot of complex features, we get the same level of classification results in the absence of optimizing classifier parameters and not use any external resources. We also demonstrated that the results of semantic relation classification can be improved by about 1% to 3% though using the weighting method based on word frequency. Compared with the existing method, the dimension of

feature vectors in our method are very low. This enables us to improve the efficiency by improving the accuracy of semantic classification and reducing the cost of calculation. And this new feature are very easy to extend and leverage, and form a new set of features combined with the common features in existing studies easily.

5.2 Future Works

Our study is related to the semantic relation classification. There are many work could be done to improve the accuracy of automatic extraction "part/material concept" and semantic relation classification, and the future works can be summarized as follows: (1) In the Associative Concept Dictionary, in addition to the "partial/material concept" data we used, many other kinds of semantic concepts that can be used as teaching data to improve the accuracy of the results to some extent. (2) When semantic relation classified, we use the nonlinear classifier to classify semantic relation only. According to the relevant research results of current neural network, for the problem of semantic relation classification, a trained deep neural network could get better results than traditional classifier. (3) Using the *substring vector* in combined with other traditional feature (excluding external data sources). For example, the position of words in sentences, syntactic information, etc. We believe that these could improve the accuracy of classification without greatly increasing the dimension of feature vectors.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my advisor Assistant Professor Dr. Chihiro Shibata for the continuous support of my Ph.D study and research, for her patience, motivation, enthusiasm, and immense knowledge. Without her assistance and guidance in every step throughout the process, this thesis would have never been accomplished.

I would like to show my deepest gratitude to Professor Dr. Kazuya Tago for his invaluable teachings and research experience. He gave me a lot of guidance on my research and teaching me how to become a qualified researcher.

Besides my advisors, I owe my deepest gratitude to the rest of my thesis committee: Associate Professor Dr. Shino Iwashita, Assistant Professor Dr. Masayuki Kikuchi and Associate Professor Dr. Yuko Osana, for their insightful comments and encouragement, but also for the hard question which inspired me to widen my research from various perspectives.

My sincere thanks also goes to Emeritus Professor Dr. Toshiyuki Kinoshita. His guidance helped me in writing of this thesis.

I am also grateful to all the members of the C.Shibata Lab and the Tago Lab for their feedback, cooperation and of course friendship.

And finally, last but by no means least, I must express my very profound gratitude to my parents and to my wife for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis.

References

[1] 潤 岡本 and 俊 石崎. 概念辞書の構築と概念空間の定量化-連想実験による概念空間の抽出-. 情報処理学会研究報告自然言語処理（*NL*）, 1999(22):81–88, 1999.

[2] 隅田 飛鳥, 吉永 直樹, and 鳥澤 健太郎. Wikipedia の記事構造からの上位下位関係抽出. 自然言語処理, 16(3), 2009.

[3] Geoffrey E. Hinton. Learning distributed representations of concepts. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, pages 1–12. Hillsdale, NJ: Erlbaum, 1986.

[4] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, 2003.

[5] Frederic Morin and Yoshua Bengio. Hierarchical probabilistic neural network language model. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pages 246–252, 2005.

[6] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *Clinical Orthopaedics and Related Research (CORR)*, abs/1301.3781, 2013.

[7] Vapnik V.N. Statistical learning theory. 1998.

[8] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344, 2014.

[9] Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. Learning semantic hierarchies via word embeddings. In *Proceedings of the 52nd Annual Meeting of the ACL*, volume 1, pages 1199–1209, 2014.

[10] G.A. Miller and Princeton University. Cognitive Science Laboratory. *Five Papers on WordNet*. Report (Princeton University. Cognitive Science Laboratory). Princeton University, Cognitive Science Laboratory, 1990.

[11] *The Structure of the EDR Electronic Dictionary*. National Institute of Information and Communications Technology, 1990.

[12] 日本電子化辞書研究所. Edr 電子化辞書使用説明書. 1990.

[13] 岡本 潤 and 石崎 俊. 概念間距離の定式化と既存電子化辞書との比較. 2001.

[14] Jun Okamoto, Kiyoko Uchiyama, and Shun Ishizaki. A contextual dynamic network model for wsd using associative concept dictionary. 2008.

[15] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. 1992.

[16] 新里 圭司 and 鳥澤 健太郎. Html 文書からの単語意味クラスの単純な自動獲得手法. 2007.

[17] Jung-Wei Fan and Carol Friedman. Word sense disambiguation via semantic type classification. In *AMIA Annual Symposium Proceedings*, page 177, 2008.

[18] Paul Nulty and Fintan Costello. General and specific paraphrases of semantic relations between nouns. *Natural Language Engineering*, 19:357–384, 2013.

[19] D Radev, J Otterbacher, and Zhu Zhang. Cross-document relationship classification for text summarization. *tangra, si, umich, edu/˜ radev/papers/progress/p1, ps (último acesso: 13/04/2009)*, 2008.

[20] Will Y Zou, Richard Socher, Daniel M Cer, and Christopher D Manning. Bilingual word embeddings for phrase-based machine translation. In *EMNLP*, pages 1393–1398, 2013.

[21] Agnese Augello, Gaetano Saccone, Salvatore Gaglio, and Giovanni Pilat. Humorist bot: Bringing computational humour in a chat-bot system. In *Complex, Intelligent and Software Intensive Systems, 2008. CISIS 2008. International Conference on*, pages 703–708. INTECH Open Access Publisher, 2008.

[22] Agnese Augello, Giovanni Pilato, Orazio Gambino, Roberto Pirrone, Salvatore Gaglio, and Vincenzo Cannella. *An Emotional Talking Head for a Humoristic Chat-bot*. INTECH Open Access Publisher, 2011.

[23] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the 47th Annual Meeting of the ACL*, volume 2, pages 1003–1011, 2009.

[24] Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting on ACL*, pages 427–434, 2005.

[25] Razvan C. Bunescu and Raymond J. Mooney. A shortest path dependency kernel for relation extraction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 724–731, 2005.

[26] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. *J. Mach. Learn. Res.*, 3:1083–1106, 2003.

[27] Yuan Chen, Man Lan, Jian Su, Zhi Min Zhou, and Yu Xu. Ecnu: Effective semantic relations classification without complicated features or multiple external corpora. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 226–229, 2010.

[28] Kateryna Tymoshenko and Claudio Giuliano. Fbk-irst: Semantic relation extraction using cyc. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 214–217, 2010.

[29] Bryan Rink and Sanda Harabagiu. Utd: Classifying semantic relations by combining lexical and semantic resources. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, 2010.

[30] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119, 2013.

[31] Tomas Mikolov, Wen tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of NAACL-HLT-2013*, pages 746–751, 2013.

[32] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó. Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, 2010.

[33] Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Conference on EMNLP*, pages 1201–1211, 2012.

[34] Kazuma Hashimoto, Makoto Miwa, Yoshimasa Tsuruoka, and Takashi Chikayama. Simple customization of recursive neural networks for semantic relation classification. In *Proceedings of the 2013 Conference on EMNLP*, pages 1372–1376, 2013.

[35] Cícero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. Classifying relations by ranking with convolutional neural networks. *CoRR*, abs/1504.06580, 2015.

[36] Kazuma Hashimoto, Pontus Stenetorp, Makoto Miwa, and Yoshimasa Tsuruoka. Task-oriented learning of word embeddings for semantic relation classification. *CoRR*, abs/1503.00095, 2015.

[37] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the ACL*, pages 384–394, 2010.

[38] Andriy Mnih and Geoffrey Hinton. Three new graphical models for statistical language modelling. In *Proceedings of the 24th International Conference on Machine Learning*, pages 641–648, 2007.

[39] Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. Recurrent neural network based language model. In *INTERSPEECH*, pages 1045–1048, 2010.

[40] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1188–1196, 2014.

[41] Yoav Goldberg and Omer Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *Clinical Orthopaedics and Related Research (CORR)*, abs/1402.3722, 2014.

[42] Michael U. Gutmann and Aapo Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *J. Mach. Learn. Res.*, pages 307–361, 2012.

[43] Tin Kam Ho. Random decision forests. In *Proceedings of the Third International Conference on Document Analysis and Recognition*, volume 1, pages 278–, 1995.

[44] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, 2001.

[45] Stephen Tratz and Eduard Hovy. Isi: Automatic classification of relations between nominals using a maximum entropy classifier. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 222–225, 2010.

[46] Aapo Hyvärinen, Patrik O. Hoyer, and Mika Inki. Topographic independent component analysis. *Neural computation*, 13:1527–1558, 2001.

[47] Quoc Le, Marc'Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg Corrado, Jeff Dean, and Andrew Ng. Building high-level features using large scale unsupervised learning. In *International Conference in Machine Learning*, 2012.

[48] Gerry Stahl, Timothy Koschmann, and Daniel Suthers. *Computer-supported collaborative learning.*, pages 479–500. Cambridge University Press, 2014.

[49] Pierre Dillenbourg, Michael J. Baker, Agnès Blaye, and Claire O'Malley. The evolution of research on collaborative learning. In Spada, E., Reiman, and P., editors, *Learning in Humans and Machine: Towards an interdisciplinary learning science.*, pages 189–211. Elsevier, Oxford, 1996.

[50] Timothy Koschmann. Understanding understanding in action. *Journal of Pragmatics*, 43(2):435 – 437, 2011.

[51] Timothy Koschmann, Gerry Stahl, and Alan Zemel. The video analyst's manifesto: (or the implications of garfinkel's policies for the development of a program of video analytic research within the learning sciences. In *Proceedings of the 6th International Conference on Learning Sciences*, ICLS '04, pages 278–285. International Society of the Learning Sciences, 2004.

[52] Michelene Chi. Quantifying qualitative analyses of verbal data: A practical guide. *Journal of the Learning Sciences*, 6(3):271–315, 1997.

[53] Kimihiko Ando, Chihiro Shibata, and Taketoshi Inaba. Towards automatic coding of collaborative learning data with deep learning technology. *Computer and Education*, 43:79–84, 2017.

[54] Kimihiko Ando, Chihiro Shibata, and Taketoshi Inaba. Analysis of collaborative learning processes by automatic coding using deep learning technology. *Computer and Education*, 43:79–84, 2017.

[55] Kimihiko Ando, Chihiro Shibata, and Taketoshi Inaba. Coding collaborative learning data automatically with deep learning methods. *JSiSE*, page 32, 2017.

[56] Taketoshi Inaba and Kimihiko Ando. Development and evaluation of cscl system for large classrooms using question-posing script. *International Journal on Advances in Software.*, 7(3 and 4):590–600, 2014.

[57] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[58] Oriol Vinyals and Quoc V. Le. A neural conversational model. *CoRR*, abs/1506.05869, 2015.

[59] Armin Weinberger and Frank Fischer. A framework to analyze argumentative knowledge construction in computer-supported collaborative learning. *Comput. Educ.*, 46(1):71–95, 2006.

[60] Sainbayar Sukhbaatar, arthur szlam, Jason Weston, and Rob Fergus. End to end memory networks. In *Advances in Neural Information Processing Systems 28*, pages 2440–2448. Curran Associates, Inc., 2015.

[61] Francesco Serafino, Gianvito Pio, Michelangelo Ceci, and Donato Malerba. Hierarchical multidimensional classification of web documents with multiwebclass. In *Discovery Science 18th International Conference, DS 2015, Banff, AB, Canada, October 4-6, 2015, Proceedings*, pages 236–250, 2015.

List of Publications

List of publications that related to this dissertation

Journals

1. Zhan Jin, Chihiro Shibata, and Toshiyuki Kinoshita. Extraction of Part / Material Concepts from Combination of Wikipedia Data and Associative Concept Dictionary. *International Journal of Machine Learning and Computing , IJMLC*, 8 2019

2. Zhan Jin, Chihiro Shibata, and Kazuya Tago. Semantic relation classification through low-dimensional distributed representations of partial word sequences. *Nonlinear Theory and Its Applications, IEICE*, 10(1):28–44, 1 2019

3. 展 靳, 公彦 安藤, 千尋 柴田, and 竹俊 稲葉. 多次元コーディングスキームに依拠した協調学習プロセスの自動コーディングの精度検証. 学習分析学 *(JASLA)*, 2:11–22, 7 2018

International Conference proceedings

1. Zhan Jin, Chihiro Shibata, and Kazuya Tago. Relation classification through substring representations using nonlinear classifiers. In *2015 International Symposium on Nonlinear Theory and its Applications (NOLTA2015)*, pages 389–392. IEICE, 8 2015

2. Zhan Jin, Chihiro Shibata, Jingtao Sun, and Kazuya Tago. On efficiency of semantic relation extraction through low-dimensional distributed representations for substrings. In *2015 IEEE International Symposium on Smart Data (Smart Data 215) in conjunction with the 17th IEEE International Conference on High Performance Computing and Communications (HPCC 2015)*, pages 1749–1754. IEEE, 6 2015

Domestic Conference proceeding

1. 展 靳, 柴田 千尋, and 田胡 和哉. 連想概念辞書および wikipedia のデータを用いた部分材料概念の抽出. In 人工知能学会全国大会論文集, volume JSAI2014, pages 4I11–4I11, 5 2014

List of publications that published during PhD. candidacy but not related to this dissertation

International Conference proceeding

1. Jingtao Sun, Zhan Jin, and Sisi Duan. A reusable adaptation language for relocation of software components on distributed systems. In *2015 International forum of IoT and Applications*, 11 2015

Domestic Conference proceedings

1. 虎遊汰 三澤, 展 靳, 千尋 柴田, and 和哉 田胡. 単語ベクトルに基づく記録文書の概念検索. In 第 *77* 回全国大会講演論文集, volume IPSJ2015, pages 187–188, 3 2015

2. 展 靳 and 田胡 和哉. クラウド向き大規模データ解析機構を用いた教育・研究文献の難易度判定. In 第 *75* 回全国大会講演論文集, volume IPSJ2013, pages 251–252, 3 2013