Text Classification of Technical Papers Using Text Segmentation

By Nguyen Hai Thien

A thesis submitted to
School of Information Science,
Japan Advanced Institute of Science and Technology,
in partial fulfillment of the requirements
for the degree of
Master of Information Science
Graduate Program in Information Science

Written under the direction of
Associate Professor Kiyoaki Shirai

September, 2012

Text Classification of Technical Papers Using Text Segmentation

By Nguyen Hai Thien (1010220)

A thesis submitted to
School of Information Science,
Japan Advanced Institute of Science and Technology,
in partial fulfillment of the requirements
for the degree of
Master of Information Science
Graduate Program in Information Science

Written under the direction of
Associate Professor Kiyoaki Shirai

and approved by
Associate Professor Kiyoaki Shirai
Professor Akira Shimazu
Associate Professor Kokolo Ikeda

August, 2012 (Submitted)

Acknowledgements

First of all, I would like to express my heartfelt thanks to my advisor **Associate Professor Kiyoaki Shirai** of **Japan Advanced Institute of Science and Technology** for his kindly guidance and supports. As my supervisor, he taught me a lot of things not only the knowledge about natural language processing but also the research methodology, developing the new idea, and solving problems.

I would like to thank **Professor Akira Shimazu**. He has given me a lot of comments to improve my research.

I would like to thank **Associate Professor Nguyen Le Minh**. He always listen to my problems and give me kind suggestion.

I received a lot of help from members in **Shimazu & Shirai Laboratory** of **Japan Advanced Institute of Science and Technology**. I owe my greats thanks to all of them

Finally, I would like to say many thanks to **FIVE-JAIST Program**. It gave me plenty of chances to improve my knowledge and research skill in Japan.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we introduce an overview of Text Classification and the motivation of our research.

1.1 Background and Motivation

Today, with the explosion of conferences and journals, there are a lot of papers in many research fields are published every year. When researchers look for a technical paper on a search engine, only papers including user's keywords will be retrieved. Some papers are not so relevant to the research topics that users want to know. It makes survey of the past researches difficult. Therefore, it will be helpful if the research topics of technical papers are automatically identified.

More specifically, the ACL Anthology is a digital archive of conference and journal papers in natural language processing and computational linguistics. ACL has a long history. In 2012, 50^{th} Annual Meeting of the Association for Computational Linguistic (ACL) was held. In the conference program, scientific papers are divided into session titles based on their research topics. However, these divisions are still subjective. Furthermore, in many cases, these divisions are incorrect and inappropriate for some papers. Therefore, if the research topics of these papers are identified, the search engine will be able to find the papers based on the research topics or filter out the papers not relevant to the research topics that the users want to know. This research aims to address this problem in order to help the survey of researches. The accurate classification of technical paper is the first crucial step for an intelligent search.

Text Classification (also known as Text Categorization) is the task to classify documents into predefined categories. It may be formalized as the task of assigning a boolean value to each pair $\langle d_i, c_j \rangle \in D \times C$, where D is a set of documents and $C = \{c_1, ..., c_{|C|}\}$ is a predefined set of categories. More formally, the task is to approximate the unknown true function $\check{\Phi} : D \times C \rightarrow \{T, F\}$ by means of a function $\Phi : D \times C \rightarrow \{T, F\}$ called the *classifier* such that $\check{\Phi}$ and Φ coincide as much as possible.

The first step that must be considered in text classification is how to represent texts. The choice of representation for texts depends on what we consider the meaning of the

textual units (focus on lexical semantics) or the combination of these units (focus on compositional semantics). A text document d_i is usually represented as a vector of *weights* $\vec{d_i} = \langle w_{1i}, \cdots, w_{|T|i} \rangle$, where T is the set of *terms* (sometimes also called *features*) which occur once or more in at least one document of the corpus. In addition, the value of weight w_{ki} is usually in the range [0,1]. Weights represent how much term t_k contributes to the semantics of document d_i. Various ways to choose important features (feature selection) and to compute term weights (feature weighting) are considered. Typical features are words in documents. This is also known as *bag of words* approach. However, using all words in the text might be not good. In addition, it may be noisy and yields worse effectiveness. Therefore, considering the structure of the documents and selecting words only in the important parts of documents would achieve better results. Scientific papers are well-organized and tend to follow a consistent sequential structure: with a Title, followed by an Abstract, Introduction, Methods, Evaluation, Conclusions and References. Among these sections, words in Title, Abstract, Introduction and Conclusion are more likely to be useful for text representation than the others.

On the other hand, the degree of generality of categories for paper classification should be considered. The fine grained categories are more useful. This leads to a better understanding the topics of papers. However, one of the problems is that the number of papers in the most categories would be small. As a result, this makes difficult to train the classifiers. In contrast, the coarse grained approaches are easy to train, but the range of categories is too broad. Therefore, the results of this approach are not very useful. In this thesis, we choose the categories as the research topics in Natural Language Processing, such as syntactic parsing, machine translation, summarization etc. After that, we try to approach fine grained topics of papers by extracting key phrases in that papers. Key phrases are defined as phrases that capture the main topics discussed in a document. Because they can give a brief and precise summary of a content of a document, we can apply them to many applications. Considering survey of past researches, the list of key phrases can quickly help researchers determine whether a given document is relevant to their interest or not.

In short, the motivation of my thesis is to help researchers conduct their survey easier and more accurate. This problem can be solved by categorizing topics of papers as text classification. However, the more fine grained categories, the more easier to understand the papers. The further step is to extract the fine grained topics by using key phrase extraction methods.

1.2 Goal of Thesis

To accomplish the motivation, there are two tasks should be considered: multi-label classification and subtopic key phrase extraction. The former task is to design an effective model which determines the categories of a given technical paper in Natural Language Processing. The categories are the research topics in Natural Language Processing, such as syntactic parsing, semantic analysis, machine translation and so on. To improve the performance, this model will consider the text segmentation. Based on the structures

of papers, our model uses only some important parts of papers for text representation such as Title, Abstract, Introduction and Conclusion. However, the categories used in this research are still broad. In addition, there are plenty of subtopics in these categories. Therefore, the latter task is to extract key phrases as subtopics of papers in order to exploit the specific subtopics of a paper. For example, a paper belonging to Machine Translation category can have subtopics such as Statistical Machine Translation, Phrase-based Statistical Machine Translation, Syntax-based Statistical Machine Translation, Alignment, and so on. Another example is that a paper belonging to Summarization category has a subtopic such as Multi-document Summarization. The aim of the latter task is to help researchers understand more deeply about the topics of the scientific papers. As in the first task, text segmentation is also considered and used in the second task.

The remaining of my thesis is organized as follows:

- Chapter 2 introduces some previous approaches on Text Classification, Multi-label Classification and Key Phrase Extraction.
- Chapter 3 describes how to construct the corpus and some statistical information about the corpus.
- Chapter 4 investigates the effectiveness of feature selections in multi-label classification through three models. We propose a novel model in multi-label classification of technical papers based on the structures of papers. We show that this model is suitable for technical paper domain.
- Chapter 5 explains two models for subtopic key phrase extraction of technical papers. Pre-processing of data and post-processing of key phrases are given and applied to these models.
- Chapter 6 assesses the experiment results of multi-label classification and subtopic key phrase extraction.
- Chapter 7 presents conclusion and future work.

Chapter 2

Background

In this chapter, we introduce some background of text classification, some previous researches on multi-label classification and key phrase extraction.

2.1 Text Classification

Text classification has a long history. Many techniques have been studied to improve the performance. The commonly used text representation is the Bag-Of-Words [1]. Not words but phrases, word sequences or N-grams [2] are sometimes used. Most of them focused on words or N-grams extracted from the whole document with feature selection or feature weighting scheme.

Some of the previous work aimed at the integration of document contents and citation structure to improve the accuracy of technical paper categorization [3] [4]. They first use the content-based classifier. Both words and phrases are used for text representation. Then the output of this classifier will be updated by using citation-based classifier. However, these researches use entire document as features in content-based classifier.

Nomoto shows that the nucleus appears at the beginning of the text, followed by any number of supplementary adjuncts [5]. Keywords for text classification are extracted only from the nucleus. We can regard segmentation of nucleus and adjuncts as a kind of text segmentation.

Larkey proposed a method to extract words only from the Title, Abstract, the first twenty lines of Summary, and the section containing the claims of novelty for a patent categorization application [6]. His method is similar to this research, but he classifies the patent documents, not technical papers.

2.2 Multi-label Classification

Many researches in text classification deal with single-label data, where training examples are associated with a single label. However, in many applications, single label classifications are not appropriate and helpful. In some applications, data may be associated with a set of labels or multi-labels in other words. For example, categorization of research

topics for technical papers, music categorization by emotions, semantic annotation for image or video etc. are examples of applications requiring multi-label classification.

There are many approaches for multi-label classification. However, they can be categorized into two groups: problem transformation and algorithm adaption [7]. The former group of methods could be based on any algorithms for single-label classification. They transform the multi-label classification task into one ore more single-label classification. On the other hand, the latter group of methods extend traditional learning algorithms to deal with multi-label data directly.

A common approach to multi-label classification is problem transformation. Single-label classifiers can be used for single-label problems, and these results will be transformed back into multi-label representations. Examples of single-label classification algorithms are Support Vector Machines, Naive Bayes, k Nearest Neighbor and so forth. The most popular way for transforming is binary approach [7], which is simple and effective. For each label, it trains a classifier using data associated with this label as positive and the others as negative. Therefore, if the training data has N labels, this method builds N classifiers. In prediction step, an instance is associated with a label if the corresponding result from the classifier is positive. Another method for transforming is label powerset [7]. It considers each different subset of labels as a single label. This method has the advantage of taking correlations among labels into account, but requires the large number of label subsets, which are associated with very few examples in most cases.

A lot of previous researches try to extend traditional algorithms to deal with multi-label data. These methods are called algorithm adaptation. AdaBoost.MH and AdaBoost.MR are two extensions of AdaBoost for multi-label data [8]. BP-MLL is an adaptation of the popular back-propagation algorithm for multi-label learning. A number of methods are based on the popular k Nearest Neighbors (kNN) lazy learning algorithm [9]. The first step in all these approaches is the same as in kNN, finding the k nearest examples. ML-kNN uses the maximum a posterior principle in order to determine the label set of the test instance, based on prior and posterior probabilities for the frequency of each label within the k nearest neighbors [10].

2.3 Key Phrase Extraction

Key Phrase Extraction is the task to identify a small set of key phrases from a document that can describe the meaning of the document. There are two existing approaches to this problem: supervised and unsupervised techniques.

Supervised methods recast this problem as a binary classification, where a model such as Naive Bayes and SVM is trained on annotated data to determine whether a given phrase is a key phrase or not [11]. Some extraction tools, e.g. KEA [12], GenEX [13], have been developed. A disadvantage of supervised approaches is that they require a lot of training data. Therefore, this method is not considered in this thesis.

In contrast, the unsupervised approaches do not need training data. The statistical information of the words such as word frequency [14], TF-IDF, the positions of the occurrences [14] [15] can be used to identify the key words in the document. Some ap-

proaches use the linguistics feature of the words, sentences and documents. The linguistics approach includes the lexical analysis, syntactic analysis, discourse analysis and so on. Other approaches for key phrase extraction have involved a number of techniques, including language modeling, graph-based ranking and clustering [16].

2.4 Discussion

The methods in [3] [4] [6] are used only for single-label classification. However, in scientific paper domain, one paper can belong to many categories. It is not appropriate to use a single-label system here. Therefore, in our system, we try to use a multi-label system to categorize papers. In addition, we conduct a key phrase extraction model to exploit the subtopics of papers. These subtopics can be regarded as fine grained categories which are difficult to train in classification.

Moreover, in scientific paper domain, documents are well-organized. In our system, we focus on only words in important parts of documents. Different from [3] [4], we introduce features that capture the positions of words, phrases in papers with respect to logical sections found in scientific discourse.

Chapter 3

Data

To develop and evaluate our proposed method of text classification, we construct a collection of technical papers associated with their correct categories. This chapter describes the way how to construct the paper collection and its statistics.

3.1 Construction of Paper Collection

We collect technical papers in proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL) from 2000-2011. Then we convert the PDF files to text files using **pdftotext** tool [1]. We remove text files converted from PDF file in image format or ones incorrectly converted. After that, we construct the list of papers with the following information: ID, Title, Authors, URL, Session and Categories. "Category" means the correct category or research topic of the paper. We will describe how to determine the category later. Table 3.1 illustrates the information of one paper in this file.

Table 3.1: Information of a Paper

Slot	**Value**
ID	P08-2051
Title	Correlation between ROUGE and Human Evaluation of Extractive Meeting Summaries
Authors	Feifan Liu; Yang Liu
URL	http://aclweb.org/anthology-new/P/P08/P08-2051.pdf
Session	Short Papers 4 (Generation/Summarization)
Categories	[Summarisation],[Evaluation methodologies]

A set of categories used in this research is shown in Table 3.2. We first refer the category list used for paper submission to the Language Resources and Evaluation Conference

[1] http://www.foolabs.com/xpdf/

(LREC). The list is modified as follows: some categories which do not appear at all or frequently are removed, some categories are added when we guess there are many papers related to these categories (research topics).

At first, we try to divide categories into 66 categories and assign each paper into these categories. In the conference program, scientific papers are divided into session titles based on their research topics. These session titles are useful information to determine the categories (research fields) of papers. Therefore, we manually construct a file which mapping a session title to categories. Table A.1 in Appendix A shows this mapping file. Some session titles such as "algorithm", "linguistic creativity", "resources" are ambiguous. In such sessions, there is no corresponding category in the mapping file. We will assign the categories of each paper in these sections manually later.

However, in many cases, categories determined by the session titles are incorrect and inappropriate for some papers. Therefore, the mapping file is just used as a hint on choosing categories. We manually check if the categories determined by the session title and the mapping file are correct, then revise categories if necessary. To ensure that each category has enough training data, the categories where the number of papers is less than 10 are removed from the collection. The total number of documents in the collection is 1,972, while the total number of categories is 38.

3.2 Statistics of Paper Collection

The right most column in Table 3.2 shows the number of papers in each category in the paper collection.

Table 3.2: Categories and the Corresponding Number of Papers

Index	**Category**	**# of papers**
1	Anaphora, Coreference	43
2	Corpus (creation, annotation, etc.)	46
3	Dialogue	102
4	Discourse annotation, representation and processing	49
5	Document Classification, Text categorisation	43
6	Evaluation methodologies	48
7	Grammar and Syntax	108
8	Information Extraction	113
9	Knowledge Discovery/Representation	24
10	Language modelling	53
11	Lexicon, lexical database	86
12	Machine Translation, SpeechToSpeech Translation	326
13	Morphology	33
14	MultiWord Expressions & Collocations	15
15	Named Entity recognition	34
16	Natural Language Generation	58
17	Ontologies	19
18	Parsing	189
19	Part of speech tagging	40
20	Phonetic Databases, Phonology	15
21	Question Answering	53
22	Semantics	71
23	Speech Recognition/Understanding	62
24	Speech Synthesis	11
25	Statistical and machine learning methods	66
26	Summarisation	78
27	Text mining	25
28	Textual Entailment and Paraphrasing	44
29	Tools, systems, applications	57
30	Word Sense Disambiguation	64
31	Chunking	14
32	Error Correction	26
33	Segmentation	32
34	Multimodal	17
35	Opinion mining / sentiment analysis / Emotion Analysis	80
36	Semantic role labeling	42
37	Information Retrieval	59
38	Psycholinguistics, Cognitive Linguistics	11

Figure 3.1 shows the distribution of papers. The horizontal axis is the categories. The vertical axis is the number of papers corresponding to categories. As you can see, "Machine Translation, SpeechToSpeech Translation" category (index 12) has the highest number of papers.

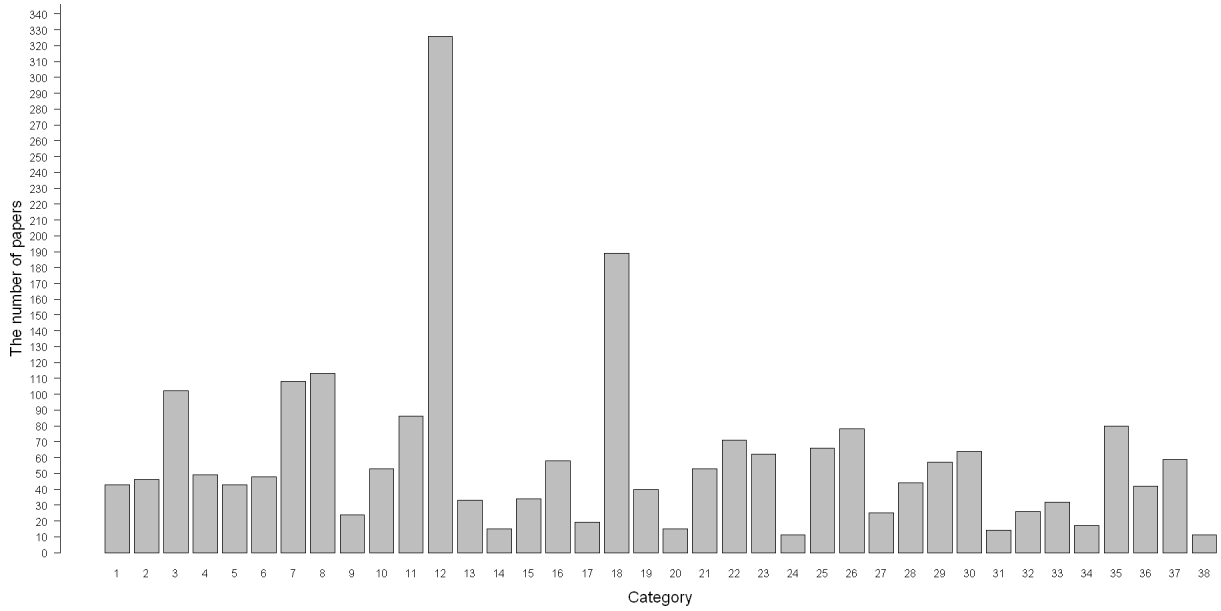

Figure 3.1: Distribution of Categories

In some applications, the number of categories of each example is small compared to the number of categories. This could be a parameter that influences the performance of the different multi-label methods. Table 3.3 summarizes this information.

Table 3.3: Distribution of Papers in Terms of Number of Correct Categories

# of categories in a paper	**# of papers**
1	1701 (86.3%)
2	259 (13.1%)
3	11 (0.6%)
4	1

Label cardinality (LC) of a dataset is the average number of categories in each paper, defined as (3.1)

$$LC = \frac{1}{m} \sum_{i=1}^{m} |Y_i| \tag{3.1}$$

,where m is the number of papers, Y_i is a set of categories of paper i.
From Table 3.3, LC of our paper collection is 1.144.

Chapter 4

Multi-label Classification of Technical Papers

In this chapter, we investigate the structure of the papers for document representation. In addition, some effectiveness of features and a novel model in multi-label classification of technical papers will be discussed. We will show that this model is suitable for technical paper domain.

4.1 Text Segmentation

Most scientific papers are subdivided into the following sections: Title, Abstract, Introduction, Methods, Experiments, Results, Conclusion and References. As you know, Abstract summarizes papers and allow the reader to judge whether papers are related to his or her own research interests. Introduction describes some background of the paper. Conclusion again summarizes papers. On the other hand, other sections discuss the details of papers, so these sections might not so helpful for classification of research topics. In addition, these sections tend to distract the information from the main topics. Therefore, in each paper, we identify the following segments: Title, Author Information, Abstract, Introduction, Conclusion and Reference. Figure 4.1 shows the standard positions of these sections in papers.

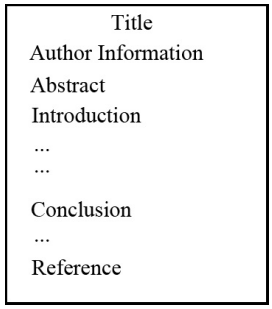

Figure 4.1: Sections of Technical Papers

Next, we will present how to identify these sections in the paper. Title sections are gotten from the database of papers shown in Chapter 3. Therefore, they are always correct. Author Information sections start from the beginning of the papers to Abstract sections. Author section contains authors' names, email addresses, affiliations and so on. Abstract, Introduction, Conclusion and Reference sections are identified by keywords in the papers. Hence, the accuracy of identification of these sections is not 100%. Algorithm 1 is used to recognize each section in papers.

Algorithm 1 Psuedocode for Identify Sections in a Paper

1: Author Information starts from the beginning of the paper.
2: Identify the beginning of Abstract section by checking the line only has a keyword "Abstract".
3: Identify the beginning of Introduction section by checking the line has a pattern "Introduction" or "1. Introduction".
4: **if** Cannot recognize the beginning of Introduction **then** Check the keywords "Motivation", "Background", "Overview" or "Objectives" in the pattern "keyword" or "1. keyword" in the line ⇒ Consider the matched line as the beginning of Introduction.
5: **end if**
6: Identify next sections by checking the lines have pattern "N. W^*" ($1 < N < 12$, W is a word, $length(W^*) < 10$, and the first characters of words are upper case).
7: Identify the beginning of Conclusion section by checking the line has a pattern "Conclusion" or "N. Conclusion" ($N_{preSection} < N < 12$, $N_{preSection}$ is the number N of previous section).
8: **if** Cannot recognize the beginning of Conclusion **then** Check the keywords "Summary" or "Discussion" in the pattern "keyword" or "N. keyword" ($N_{preSection} < N < 12$) in the line ⇒ Consider the matched line as the beginning of Conclusion.
9: **end if**
10: Identify the beginning of Reference section by checking the line has the keywords "Reference", "Bibliography" in the pattern "keyword" or "N. keyword" ($N_{preSection} < N < 12$).
11: The end of each section is also regarded as the beginning of the next section.

To see the overview of the number of text segmentation, Table 4.1 summarizes the number of unrecognized and recognized sections in the paper collection. Text segments are identified in most papers. Conclusion is the most difficult for identification. However, only 6%($\frac{120}{120+1852}$) papers are failed to extract. When we checked a small subset of the collection, all recognized segments are correct. Therefore, we assumed that the accuracy of text segmentation would be high. One of the main reasons for not being able to recognize some sections is that some papers do not actually contain these sections. Another reason is that the converted corresponding keywords from PDF files to text files are incorrect.

Table 4.1: The Number of Unrecognized and Recognized Sections

Sections	**# of unrecognized sections**	**# of recognized section**
Abstract	3	1969
Introduction	28	1944
Conclusion	120	1852
Reference	1	1971

4.2 Feature

Document representation is one of the most important issues in text processing, especially in text categorization. As usual, we represent a document as a feature vector. Basically, words in the paper are used as features. Stop words and numbers are removed from the features, since they are ineffective for text classification. Furthermore, words in Author Information and Reference are also removed. Then, the content words are lemmatized by the Stanford CoreNLP [1]. All lemmatized forms of content words are used as features.

In addition to the above bag-of-words features, we propose new types of feature derived from the title of the paper. Words in the title are the most important features for paper classification. These words specify topics of papers. However, only some words in titles may be effective features. Some other words represent the details of the main theme of the paper and usually not represent the topic of the paper. If we use all words in the title, some noisy words may be extracted and they would give negative impact for classification. In this thesis, 'Title Bi-Gram' and 'Title SigNoun' are proposed to overcome this weakness.

Words in title sections do not usually make up a sentence. Almost titles are just phrases. In addition, the main words in these phrases tend to appear in Noun Phrases (NPs). While words in phrases other than NPs are more likely to be detail words. Considering above, 'Title Bi-Gram' is defined as bi-gram in noun phrases in the title. The motivation of 'Title Bi-Gram' feature is that the title usually describes topics in the noun phrases. Furthermore, the topics are often represented by not a single word but a phrase. Therefore, this method tries to capture these phrases in titles.

Another title feature is 'Title SigNoun', which is defined as significant nouns in the title. Two types of significant nouns are used as this feature. One is nouns in a head NP. Here 'head NP' stands for a noun phrase including the head of the whole title. The other is nouns in prepositional phrases (PPs). This feature is represented in the form of 'p+n', where n and p is a noun in PP and a head preposition of PP, respectively. The motivation of 'Title SigNoun' feature is that not only the nouns in the head NP but also in some cases the words in the prepositional phrase describe topics of papers. For example, a prepositional phrase "with bilingual lexicon" is not useful because the nouns in this phrase might not help to identify topics of papers. In contrast, a prepositional phrase "for information retrieval" is very useful, since 'for' may represent the purpose of the paper. So this phrase gives an information that the paper tends to belong to

[1] `http://nlp.stanford.edu/software/corenlp.shtml`

"Information Retrieval" category. Therefore, this method tries to use the combination of the noun with the preposition, such as 'for+retrieval', to distinguish effective and ineffective prepositional phrases.

The Stanford CoreNLP is used to get the parse tree of the title for Title Bi-Gram and Title SigNoun features. In addition, dependency relations are also extracted for Title SigNoun feature by Stanford CoreNLP. Dependency relation represents relationships between pairs of words in the sentence with their relation type. For example, if the title is "Annotating and Recognising Named Entities in Clinical Notes", Stanford parser outputs Figure 4.2 and Figure 4.3 as the parse tree and dependency relation, respectively. Then 'Named Entities' and 'Clinical Notes' are extracted as Title Bi-Gram, while 'Named', 'Entities' and 'in+Notes' are extracted as Title SigNoun feature.

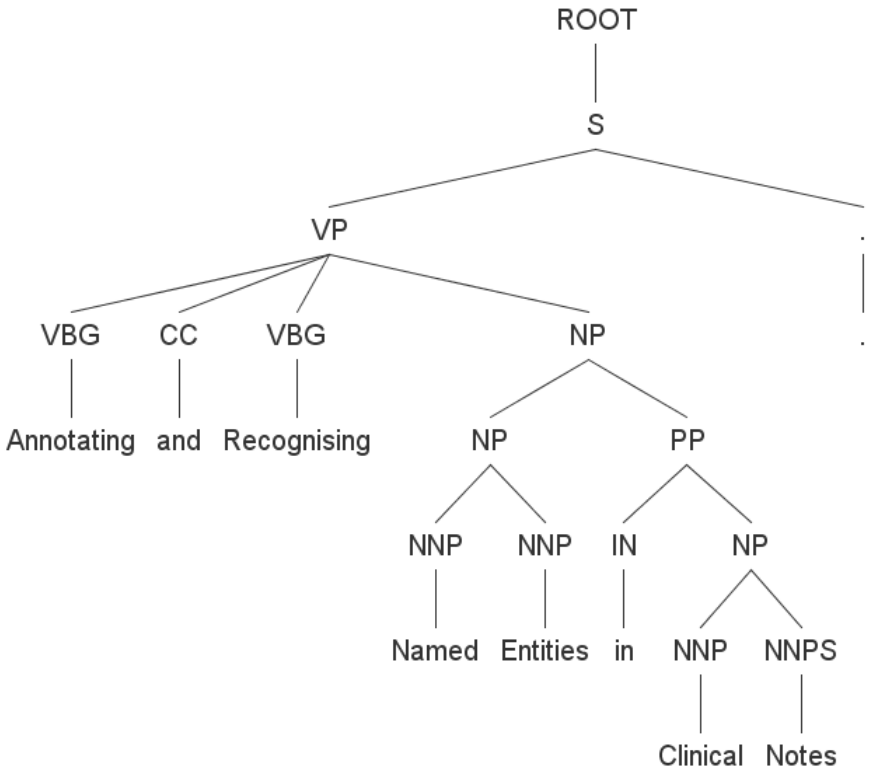

Figure 4.2: Sample Parse Tree of a Title

```
<parse>
   (ROOT (S (VP (VBG Annotating) (CC and) (VBG Recognising)
        (NP (NP (VBN Named) (NNS Entities)) (PP (IN in)
        (NP (JJ Clinical) (NNS Notes)))))))
</parse>
<basic-dependencies>
  <dep type="cc">
    <governor idx="1">Annotating</governor>
    <dependent idx="2">and</dependent>
  </dep>
  <dep type="conj">
    <governor idx="1">Annotating</governor>
    <dependent idx="3">Recognising</dependent>
  </dep>
  <dep type="amod">
    <governor idx="5">Entities</governor>
    <dependent idx="4">Named</dependent>
  </dep>
  <dep type="dobj">
    <governor idx="1">Annotating</governor>
    <dependent idx="5">Entities</dependent>
  </dep>
  <dep type="prep">
    <governor idx="5">Entities</governor>
    <dependent idx="6">in</dependent>
  </dep>
  <dep type="amod">
    <governor idx="8">Notes</governor>
    <dependent idx="7">Clinical</dependent>
  </dep>
  <dep type="pobj">
    <governor idx="6">in</governor>
    <dependent idx="8">Notes</dependent>
  </dep>
</basic-dependencies>
```

Figure 4.3: Sample Dependency Relation of a Title Section

The algorithm to extract Title Bi-Gram feature is that first any NPs are identified from the parse tree, then any two consecutive nouns in NPs are extracted. To extract Title SigNoun feature, first the head NP of the title is identified from the parse tree. Algorithm 2 shown below is used to accomplish this step. Then any nouns in the head NP are extracted as Title SigNoun features. Furthermore, for all nouns n in prepositional phrases, we check if n has a dependency relation with a preposition p where the relation type is 'pobj' (object of a preposition). Then all p+n are extracted as another type of Title SigNoun feature.

Algorithm 2 Psuedocode for Retrieving Head Noun Phrase in a Title

1: Run Stanford CoreNLP to get the parse tree of the sentence.
2: Find the top NP from top to bottom of parse tree.
3: **if** Exist top NP **then** Find the leaf NP in this top NP branch. (Ignore the other branches). ⇒ Consider this leaf NP as a head NP.
4: **else** Try finding top VP. Find the leaf NP in this top VP branch. (Ignore the other branches). ⇒ Consider this leaf NP as a head NP.
5: **end if**

4.3 Feature Selection

We propose a method of feature selection based on the structure or segments of the paper. As discussed earlier, words in Title, Abstract, Introduction and Conclusion may be useful for classification of research topics, while words in other segments may be ineffective or noisy features. In our feature selection algorithm, first text segmentats are identified by the method described in Section 4.1. Then only words in useful segments such as Title, Abstract, Introduction and Conclusion are selected as features.

Considering feature types and feature selection based on the text segmentation, this thesis uses four feature sets as follows:

1. The whole content of paper: all of the words in the papers will be selected as features.

2. Title, Abstract, Introduction, Conclusion: only words in these parts of the papers will be selected as features.

3. Title, Abstract, Introduction, Conclusion + Title Bi-Gram: words in these parts as well as Title Bi-gram are used as features.

4. Title, Abstract, Introduction, Conclusion + Title SigNoun: words in these parts as well as Title SigNoun are used as features.

In Text Classification, the high dimensionality of the feature space may be problematic. Therefore, dimensionality reduction is usually used as a kind of feature selection. It can be defined as a method to reduce the dimensionality of the vector space from $|T|$ to $|T'|$, where $|T'| \ll |T|$. The set T' is called the reduced term set. One of the benefits is that it tends to avoid over-fitting. Document Frequency can be simply and effectively used for dimensionality reduction. The idea to use document frequency is that the most valuable terms for text classification are those that occur many documents in the collection. In our paper collection, the set of terms that occur only one document in the collection is large. Thus, to reduce the feature spaces, the features whose document frequency is less than 2 are removed. Note that dimensionality reduction based on document frequency and feature selection based on the text segments can be used simultaneously.

4.4 Feature Weighting

Feature Weighting is a method assigning appropriate weights to the features in order to reflect how important features are in documents. In this research, we conducted experiments with two traditional feature weighting: Binary Weighting and TF-IDF.

1. **Binary Weighting:** is the simplest method for document representation. Each term weight is 0 or 1, 0 means the term is not present and 1 means the term is present in the document.
$$w_{ij} = \begin{cases} 1 & \text{if } t_i \in d_j \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

2. **TF-IDF:** combines **Term Frequency** and **Inverse Document Frequency**. TF-IDF of $word_i$ in $document_j$ is:
$$w_{ij} = tf_{ij} * (log_{10}\frac{M}{DF_i} + 1) \tag{4.2}$$
tf_{ij}: frequency of $term_i$ in $document_j$.
DF_i: document frequency.
M: the total number of documents in a corpus.

4.5 Models

As pointed out in Chapter 2, we can classify the existing methods for multi-label classification into two main approaches: problem transformation and algorithm adaption. To compare the advantages and evaluate the performance, this research investigates one method in each group. We choose ML-kNN described in (4.5.1) as algorithm adaptation and binary approach described in (4.5.2) as problem transformation. After that, a novel model is constructed based on the structure of papers.

For the formal description of the methods, we will use $C = \{c_j : j = 1 \dots |C|\}$ to denote the finite set of labels in a multi-label learning task and $D = \{(x_i, Y_i), i = 1 \dots m\}$ to denote a set of multi-label training examples, where x_i is the feature vector of a document and $Y_i \subseteq C$ the set of labels of the i^{th} document.

4.5.1 ML-kNN: Multi-label Learning K-Nearest Neighbors

ML-kNN [10] which is derived from the traditional k-Nearest Neighbor algorithm is a multi-label lazy learning approach. In detail, for each unseen instance, its k nearest neighbors in the training set are firstly identified. After that, the label set for the unseen instance are determined by using statistical information obtained from the label sets of its neighboring instances such as maximum a posterior principle.

More specifically, given an instance x belonging to label set $Y \subseteq C$, let $\vec{y}_x = \{l_j : j = 1 \dots |C|\}$ denote the category vector for instance x, where $l_j = 1$ if $l_j \in Y$. Moreover,

let $N(x)$ denote the set of K nearest neighbors of x. A membership counting vector $\overrightarrow{MC}_x = \{mc_j : j = 1 \dots |C|\}$ is counted based on the label sets of these neighbors.

$$\overrightarrow{MC}_x = \sum_{a \in N(x)} \vec{y}_a \tag{4.3}$$

From Equation (4.3), mc_j is the number of neighbors of x belonging to the i^{th} class. For each test instance t, its K nearest neighbors $N(t)$ is identified in the training set. Let $H_1^{l_j}$, and $H_0^{l_j}$ be the events that instance t has and does not have the label l_j, respectively. $E_n^{l_j}$ denotes that there are n neighbors of instance t which have label l_j. The final aim is to calculate the category vector $\vec{y}_t$ for instance t by using maximum a posterior principle:

$$\vec{y}_t(l_j) = argmax_{b \in \{0,1\}} P(H_b^{l_j} | E^{l_j}_{\overrightarrow{MC}_t}) \tag{4.4}$$

Using the Bayesian rule, Equation (4.4) can be rewritten as:

$$\begin{aligned} \vec{y}_t(l_j) &= argmax_{b \in \{0,1\}} \frac{P(H_b^{l_j}) P(E^{l_j}_{\overrightarrow{MC}_t} | H_b^{l_j})}{P(E^{l_j}_{\overrightarrow{MC}_t})} \\ &= argmax_{b \in \{0,1\}} P(H_b^{l_j}) P(E^{l_j}_{\overrightarrow{MC}_t} | H_b^{l_j}) \end{aligned} \tag{4.5}$$

As we pointed out in Equation (4.5), all the necessary information to determine the category vector $\vec{y}_t$ is the prior probabilities $P(H_b^{l_j})$ and the posterior probabilities $P(E^{l_j}_{\overrightarrow{MC}_t} | H_b^{l_j})$. These probabilities are directly estimated from the training set. However, in the prediction outcome, it is possible that a paper is not associated with any category. This happens when all l_j in predicted category vector $\vec{y}_t$ are equal to 0. In such cases, we choose one category c_j whose probability is higher than the other categories as predicted categories for these papers.

4.5.2 Binary Approach

Binary Approach [7] is a popular problem transformation method that learns $|C|$ binary classifiers, one for each different label in C. It transforms the original data set into $|C|$ data sets $D_{c_j} (j = 1 \dots |C|)$ that contain all examples of the original data set, where labeled as positive if the label set of the original example contained c_j and negative otherwise. Table 4.3 shows the four data sets that are constructed by Binary Approach when applied to the data set of Table 4.2

Table 4.2: Example of a Multi-label Data Set

Example	**Attributes**	**Label set**
1	x_1	$\{c_1, c_4\}$
2	x_2	$\{c_3, c_4\}$
3	x_3	$\{c_1\}$
4	x_4	$\{c_2, c_3, c_4\}$

Table 4.3: Data Sets Produced by the Binary Approach Method

Ex.	Label	Ex.	Label	Ex.	Label	Ex.	Label
1	c_1	1	$\neg c_2$	1	$\neg c_3$	1	c_4
2	$\neg c_1$	2	$\neg c_2$	2	c_3	2	c_4
3	c_1	3	$\neg c_2$	3	$\neg c_3$	3	$\neg c_4$
4	$\neg c_1$	4	c_2	4	c_3	4	c_4

To train each binary classifier, any kinds of traditional classifier can be utilized. Support Vector Machine (SVM) is used as a binary classifier in this model. More specific, we use the tool called LibSVM to conduct the experiments. There are four common kernel types supported by LibSVM: linear, polynomial, radial basis and sigmoid. However, the number of features in text classification is very large. Complex kernel requires much computational time with a large number of features. Moreover, using linear kernel, the classifier can quickly train the data. Therefore, the kernel we chose in this model is a linear kernel. To make further decision based on the output, we configure LibSVM to get posterior probability $P(class|sample)$. This probability is proportional to the perpendicular distance of the point which represents for the paper from the separate hyper-plane. For the classification of a new instance, Binary Approach outputs the union of the labels c_j that are positively predicted by the classifiers. However, in the prediction outcome, it is possible that a paper is not associated with any categories. In such cases, we assign one category having the maximum posterior probability among $|C|$ classifiers for these papers.

4.5.3 Back-off Model

Based on the structure of papers, we propose a new model derived from the above Binary Approach. To improve the precision, only categories with high posterior probability from different perspectives are selected. The perspectives are Binary Approach methods with different feature sets. As we guess, titles of papers are the concise information about the topics of papers. Therefore, in the first step of this model, only words in Title are selected as the feature set. Using Binary Approach with this feature selection is carried out. In the output, papers belonging to categories with high probability are removed from test set. This procedure is repeated in next steps with other feature selections: Abstract, Introduction and Conclusion. If there are some papers not belonging to any category after running previous steps, the final step will be conducted. In this step, all categories and their probabilities outputted from previous steps are put into Max Probability function. This function first chooses the categories in any steps having the probability greater than 0.5. If all categories of papers in all steps are lower than 0.5, the category having the maximum probability among these steps will be assigned to that paper.

Algorithm 3 shows Max Probability function in the final step of this model.

Algorithm 3 Psuedocode for the Max Probability function

```
N: the number of papers where the categories are not determined.
S: the number of steps in this model.
for i=1 to N do
    for j=1 to S do Assign categories having probability greater than 0.5 to current paper i.
    end for
    if Paper i still does not belong to any category. then Assign category having highest probability among S steps.
    end if
end for
```

For deeper understanding of this system, Figure 4.4 describes the steps when one paper is inputted. In this figure, n denotes the number of categories. P_{ik} denotes the posterior probability that this paper belongs to $category_i$ in $model_k$. At first, model 1, which uses words only in Title section as feature selection, judges categories for the paper. The results of model 1 are posterior probabilities associated with categories $[(c_1, p_{11}), (c_1, p_{21}), \ldots, (c_n, p_{n1})]$. If there is at least one p_{i1} greater than a threshold T_1, the categories c_i will be chosen as categories of this paper and our system goes on other papers. Otherwise, this paper will go through model 2, which used words in Title and Abstract sections. Similarly, this model outputs $[(c_1, p_{12}), (c_1, p_{22}), \ldots, (c_n, p_{n2})]$ as results. Only categories which have probabilities p_{i2} greater than a threshold T_2 will be chosen as this paper's categories. Otherwise, the next two models with thresholds T_3, T_4 are used with the same procedure. If there is no probability p_{i4} greater than a threshold T_4, the max probability function will be used to choose at least one category for this paper.

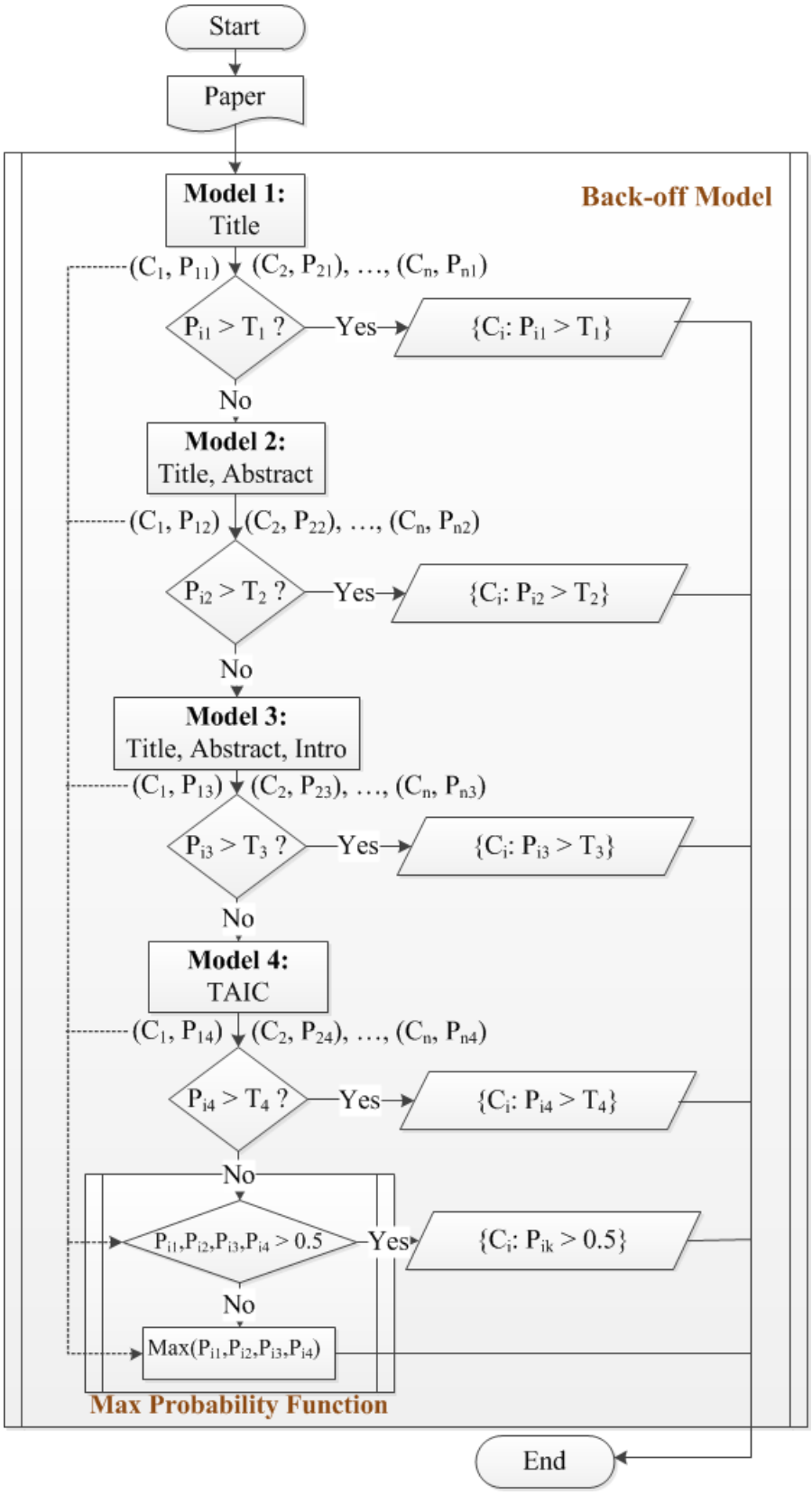

Figure 4.4: Back-off Model

Because this model will classify easy instances first and then move on harder ones in the next steps, the degree of certainty drops as the degree of difficulty increases. Therefore, the threshold (the degree of certainty) at each step will be smaller than the previous steps. We investigate several sets of thresholds for four models in the experiments in Chapter 6.

Chapter 5

Subtopic Key Phrase Extraction of Technical Papers

In this chapter, we will focus on extracting the specific subtopics of papers in order to identify more fine grained topics.

5.1 Preprocessing

The important words representing the topics of papers are usually adjective or noun. Therefore, all of words in papers are part-of-speech (POS) tagged by running Stanford CoreNLP as preprocessing.

To saving space and making reference effectively, authors often introduce acronyms for phrases that are used many times in papers. Moreover, definitions of acronyms are usually found before the acronym between parentheses. For example, to reuse "Statistical Machine Translation" phrase, authors can give an acronym following this phrase as "Statistical Machine Translation (SMT)". Therefore, we propose a simple algorithm to find all acronyms and their corresponding definitions.

Algorithm 4 Psuedocode for Acronym & Definition Detection

```
Retrieve all the upper capital terms T_1 ... T_N within parentheses () in paper
for i=1 to N do
    if The preceding consequence words coincide with corresponding characters in T_i then
Consider T_i as an Acronym.
The preceding consequence words are regarded as the definition for T_i
    end if
end for
```

After finding the acronyms and definitions, each acronym will be replaced by its definition in papers. One of the motivation of finding and replacing acronyms by their

definitions is that it makes the meaning of acronyms more clearly. In addition, some words in acronyms appear many times in documents, so it increases the frequency of each word in that phrase. As explained later, frequency of words are used to extract key phrases. Finally, POSs will be tagged more precisely for sentences contain Acronyms.

5.2 Models

To extract the subtopics of papers, we investigate two models based on graph-based ranking.

5.2.1 TextRank

In TextRank algorithm [17], a text is represented by a graph. Each vertex corresponds to a word in papers. If two words co-occur within a window of W words, there will be an edge connecting two corresponding vertices. In the traditional TextRank algorithm, edges are unweighted. However, the number of times two words co-occur within a window is important information. Therefore, a weight w_{ij} is assigned to the edge, and its value is the number of times that two words co-occur within a window.

The score of each vertex, which reflect its importance, is computed as Equation (5.1)

$$S(v_i) = (1-d) + d * \sum_{v_j \in Adj(v_i)} \frac{w_{ji}}{\sum_{v_k \in Adj(v_j)} w_{jk}} S(v_j) \tag{5.1}$$

where $Adj(v_i)$ denotes v_i's neighbors and d is the damping factor set to 0.85. Equation (5.1) is a recursive formula. The score for vertex v_i is initialized with a value of 1 and is computed in an iterative manner until convergence. After convergence, several words that correspond to the top ranked vertices are used to form key phrases for papers. All of these words are selected as potential keywords and marked in the text. After that, sequences of adjacent keywords are collapsed into a key phrase. For example, in the text "focus on statistical machine translation method", if *statistical*, *machine* and *translation* are selected as potential keywords by TextRank, as they are adjacent, they are collapsed into a phrase "statistical machine translation".

5.2.2 SingleRank

SingleRank [18] is similar to TextRank with three major differences. First, each edge in a TextRank graph is unweighted, while each edge in a SingleRank has a weight equal to the number of times the two corresponding word co-occur. However, as noted before, we also change TextRank from unweighted to weighted graph. Second, in TextRank, only the word that corresponds to the top ranked vertices can be used to form key phrases. In SingleRank, the authors do not filter out any low-scored vertices. Third, the way forming key phrases in TextRank is just to collapse adjacent words corresponding to the top ranked vertices. In SingleRank, it first selects candidate key phrases which are any longest-matching sequence of nouns and adjectives in the document. It then calculates

the scores of candidate key phrases by summing the scores of their constituent words. After that, it outputs the N highest scored candidates as the key phrases for the text. For instance, in the text "focus on statistical machine translation method", SingleRank extracts "statistical machine translation method" as a candidate key phrase. The score of this phrase is calculated by summing the scores of words in this phrase.

5.3 Postprocessing of Key Phrase Extraction

The results of TextRank and SingleRank are the key phrases of papers. However, these results are overlapped and some key phrases are not subtopics of papers. Therefore, we carried out the postprocessing step to filter out the overlapped and incorrect key phrases. The key phrases tend to occur at the beginning of the paper. As a result, the information about which sections key phrases appear in is the important clue to filter out the incorrect key phrases. Key phrases appear in Title sections are more likely to be subtopics of papers. In contrast, if key phrases appear only in Conclusion sections, these key phrases tend not to be subtopics. Another helpful information is frequency of key phrases. The higher frequency, the more likely key phrases are subtopics.

To avoid overlap results, if a key phrase is considered as a subtopic, we choose the longest key phrase containing this key phrase.

The following list is some heuristic rules of postprocessing key phrases:

Heuristic rules:

1. Remove key phrases just appear in **Conclusion** sections.
2. If it appears in **Title** sections, then just choose their longest union key phrases.
3. If $frequencycount > 1$ and $\frac{frequencycount}{\sum_{i \in K} frequencycount_i} > 0.3$ (K is the set of key phrases extracted by TextRank or SingleRank), then choose their longest union key phrases.
4. If $frequencycount >= 5$, then choose their longest union key phrases.

In above rules, "longest union key phrase" means the longest sequence of words which subsumes all given key phrases.

To illustrate this process, we choose a paper which has a title "Syntax-based Statistical Machine Translation using Tree Automata and Tree Transducers" which belongs to "Machine Translation, SpeechToSpeech Translation" category. Table 5.1 shows the key phrases extracted by TextRank algorithm and some statistical information about them.

Table 5.1: Sample Extracted Key Phrases by TextRank and Their Statistical Information

Key Phrases	Frequency	First Position	Title	Abstract	Intro	Conclusion
statistical machine translation	9	2	1	1	7	0
language model	1	47	0	1	0	0
machine translation	13	3	1	1	11	0
statistical machine translation system	4	177	0	0	4	0
translation model	3	274	0	0	3	0

In Table 5.1, Frequency column is the number of times that key phrase appears in this paper. First Position column is the position of key phrase which first appears in this paper. Title, Abstract, Intro, Conclusion columns count times that key phrase appears in Title, Abstract, Introduction and Conclusion section, respectively.

First, all of these key phrases appear in sections rather than Conclusion, so no phrase is removed. Second, there are two phrases appearing in Title section: "statistical machine translation" and "machine translation". The longest union key phrase of these two phrases is "statistical machine translation system". Therefore, we choose this phrase as a subtopic of this paper. Next, only "machine translation" phrase satisfies the 3^{rd} rule. This phrase has already extracted by the 2^{rd} rule. Finally, there are two phrases satisfy the 4^{rd} rule, but these phrases are also extracted by the 2^{rd} rule. In short, the result of postprocessing process of key phrase extraction from Table 5.1 is "statistical machine translation system". We consider this result as subtopic of this paper.

Chapter 6

Evaluation

In this chapter, we will carry out some experiments and evaluate the performance for two tasks: Multi-label Classification and Subtopic Key Phrase Extraction.

6.1 Evaluation of Multi-label Classification

6.1.1 Experiment Setup

Cross Validation

The collection of the paper described in Chapter 3 is divided into 10 parts, and we use cross validation to evaluate for each method.

ML-kNN

MULAN [19] is an open-source java library for learning from multi-label dataset. The library includes a variety of state-of-the-art algorithms for performing major multi-label learning tasks: "Classification", "Ranking" and "Classification and ranking". In addition, it also offers an evaluation framework that calculates a large variety of evaluation measures through hold-out evaluation and cross-validation. We used the MULAN library for ML-kNN algorithm.

Binary Approach

When using Binary Approach, we learn $|C|$ binary classifiers, one for each different label in C. For each binary classifier, LibSVM [20] is used to classify categories of papers.

Back-off Model

In each step of Back-off Model, Binary Approach with LibSVM for single-label classification is used. After running the two previous methods, we found that TF-IDF is a better feature weighting method than Binary Weighting. Therefore, in this model, we use only TF-IDF method for feature weighting. To investigate the effectiveness of threshold $T_i$in

each step, we vary these parameters and compare the results. Furthermore, combining Title Bi-Gram and Title SigNoun tends to give better results in Binary Approach. Because Back-off model is based on Binary Approach, we try to use these features. We conduct four models as follows:

- Back-off Model 1:
  1. Title + Title SigNoun: with Threshold T_1.
  2. Title + Title SigNoun + Abstract: with Threshold T_2.
  3. Title + Title SigNoun + Abstract + Intro: with Threshold T_3.
  4. Title + Title SigNoun + Abstract + Intro + Conclusion: with Threshold T_4.
- Back-off Model 2:
  1. Title + Title SigNoun + DF: with Threshold T_1.
  2. Title + Title SigNoun + Abstract + DF: with Threshold T_2.
  3. Title + Title SigNoun + Abstract + Intro + DF: with Threshold T_3.
  4. Title + Title SigNoun + Abstract + Intro + Conclusion + DF: with Threshold T_4.
- Back-off Model 3:
  1. Title + Title Bi-Gram: with Threshold T_1.
  2. Title + Title Bi-Gram + Abstract: with Threshold T_2.
  3. Title + Title Bi-Gram + Abstract + Intro: with Threshold T_3.
  4. Title + Title Bi-Gram + Abstract + Intro + Conclusion: with Threshold T_4.
- Back-off Model 4:
  1. Title + Title Bi-Gram + DF: with Threshold T_1.
  2. Title + Title Bi-Gram + Abstract + DF: with Threshold T_2.
  3. Title + Title Bi-Gram + Abstract + Intro + DF: with Threshold T_3.
  4. Title + Title Bi-Gram + Abstract + Intro + Conclusion + DF: with Threshold T_4.

To sum up, Model 1 and 2 use Title SigNoun feature, while Model 3 and Model 4 use Title Bi-Gram feature. Furthermore, dimensionality reduction by Document Frequency is performed in Model 2 and 4, but not in Model 1 and 3.

Metrics

Some of the measures are calculated based on the differences of the actual and the predicted sets of labels over all examples of the evaluation data set. Others decompose the evaluation process into separate evaluations for each label, which they subsequently average over all labels. We call the former instance-based and the latter category-based evaluation measures.

Instance-based Metrics

There are 4 kinds of metrics: Exact Match Ratio (EMR), Accuracy, Precision and Recall. Let us suppose that the test data with gold multi-label is (x_i, Y_i) $(i = 1..m)$, and Z_i is a set of labels that are predicted by the classifier

$$\textbf{Exact Match Ratio} = \frac{1}{m}\sum_{i=1}^{m} I[Z_i = Y_i] \tag{6.1}$$

where I(true) = 1 and I(false) = 0. This is a very strict evaluation measure as it requires the predicted set of labels to be exactly matched with the true set of labels. While the Exact Match Ratio does not care about the partial matching, the Accuracy metric calculates the partial matching in each paper.

$$\textbf{Accuracy} = \frac{1}{m}\sum_{i=1}^{m} \frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|} \tag{6.2}$$

Precision and Recall are also used as evaluation criteria.

$$\textbf{Precision} = \frac{1}{m}\sum_{i=1}^{m} \frac{|Y_i \cap Z_i|}{|Z_i|} \tag{6.3}$$

$$\textbf{Recall} = \frac{1}{m}\sum_{i=1}^{m} \frac{|Y_i \cap Z_i|}{|Y_i|} \tag{6.4}$$

Category-based Metrics

Any known measures for binary evaluation can be used here, such as precision, recall, and F-measure. The calculation of these measures for all labels can be achieved using two averaging operations, called macro-averaging and micro-averaging.

Binary evaluation measures are calculated based on the number of true positives (tp), true negatives (tn), false positives (fp) and false negatives (fn). True positives is the number of documents correctly categorized for the positive samples. True negatives is the number of documents correctly rejected for the negative samples. False positives is the number of documents incorrectly categorized for the positive samples. False negatives is the number of documents incorrectly rejected for the negative samples. These can be summarized in the contingency table in Table 6.1.

Table 6.1: The Contingency Table

		Gold judgments	
		YES	**NO**
Classifier judgments	**YES**	tp	fp
	NO	fn	tn

Let tp_c, fp_c, tn_c, fn_c be the number of true positives, false positives, true negatives and false negatives of a binary classifier for a label c. MicroPrecision, MicroRecall and MicroF are calculated as follows:

$$\textbf{MicroPrecision} = \frac{\sum_{c \in C} tp_c}{\sum_{c \in C} (tp_c + fp_c)} \tag{6.5}$$

$$\textbf{MicroRecall} = \frac{\sum_{c \in C} tp_c}{\sum_{c \in C} (tp_c + fn_c)} \tag{6.6}$$

$$\textbf{MicroF} = \frac{2 * MicroPrecision * MicroRecall}{MicroPrecision + MicroRecall} \tag{6.7}$$

MacroPrecision, MacroRecall and MacroF are calculated by first evaluating predicted categories locally for each category, and then globally by averaging over the results of the different categories:

$$\textbf{MacroPrecision} = \frac{\sum_{c \in C} \frac{tp_c}{tp_c + fp_c}}{|C|} \tag{6.8}$$

$$\textbf{MacroRecall} = \frac{\sum_{c \in C} \frac{tp_c}{tp_c + fn_c}}{|C|} \tag{6.9}$$

$$\textbf{MacroF} = \frac{\sum_{c \in C} \frac{2 * tp_c}{2 * tp_c + fp_c + fn_c}}{|C|} \tag{6.10}$$

6.1.2 Results

ML-kNN: Multi-label Learning K-Nearest Neighbors

Table 6.2 reveals results of ML-kNN with several feature sets. The values of the best system for each evaluation metrics is represented in bold. From Table 6.2, the maximum value of Exact Match Ratio (EMR) is 42.54% and Micro-F is 47.85% by using words only in Title, Abstract, Introduction, Conclusion and combining Title Bi-Gram feature method

with dimensional reduction for feature selection and TF-IDF for feature weighting. The highest Macro-F is 40.33% by using words only in Title, Abstract, Introduction, Conclusion for feature selection with dimensional reduction, and TF-IDF for feature weighting.

Table 6.2: Results of ML-kNN with K = 100

Term Selection	**TW**	**Metrics**									
		Instance-based				**Category-based**					
		EMR	**A**	**P**	**R**	**Mi-P**	**Mi-R**	**Mi-F**	**Ma-P**	**Ma-R**	**Ma-F**
All	BW[1]	39.91	44.32	48.63	44.55	48.73	42.90	45.63	42.72	26.02	37.39
	TF-IDF	34.23	37.68	41.15	37.76	41.18	36.08	38.46	33.90	20.70	29.26
All + DF[2]	BW	41.68	46.06	50.43	46.24	50.48	44.47	47.28	46.29	26.63	38.08
	TF-IDF	40.41	44.27	47.99	44.54	48.04	42.45	45.07	43.15	28.21	35.83
TAIC[3]	BW	36.81	40.94	45.00	41.12	45.06	39.61	42.16	42.75	22.75	33.67
	TF-IDF	38.44	42.40	46.30	42.59	46.37	40.84	43.43	44.29	27.32	36.48
TAIC + DF	BW	37.98	42.14	46.35	42.23	46.33	40.62	43.29	44.80	24.13	35.23
	TF-IDF	41.43	45.86	50.02	46.27	50.23	44.57	47.22	47.82	29.75	**40.33**
TAIC + Title SigNoun	TF-IDF	39.04	42.62	46.22	42.73	46.30	40.74	43.34	41.02	26.16	35.35
TAIC + Title SigNoun + DF	TF-IDF	41.48	45.96	50.29	46.27	50.39	44.56	47.30	46.73	29.00	39.24
TAIC + Title Bi-Gram	TF-IDF	39.30	43.33	47.29	43.56	47.34	41.77	44.38	43.01	26.60	36.40
TAIC + Title Bi-Gram + DF	TF-IDF	**42.54**	**46.69**	**50.59**	**47.05**	**50.87**	**45.17**	**47.85**	**49.03**	**30.49**	40.09

To assess the effectiveness of feature selection based on text segmentation, we mainly compare Exact Match Ratio, Micro-F and Macro-F metrics. Figure 6.1 and 6.2 compare results of All and TAIC or All + DF and TAIC + DF with Binary Weighting and TF-IDF Weighting. In the Binary Weighting method, feature selection by text segmentation gives worse results than use of all contents. Exact Match Ratio drops 3.1% from All and drops 3.7% when using dimensionality reduction by Document Frequency. Similarly, Micro-F drops 3.47% and 3.99%. Macro-F drops 3.72% and 2.85%.

[1]BW: feature weighting by Binary Weighting method

[2]DF: dimensionality reduction by Document Frequency

[3]TAIC: Title, Abstract, Introduction and Conclusion

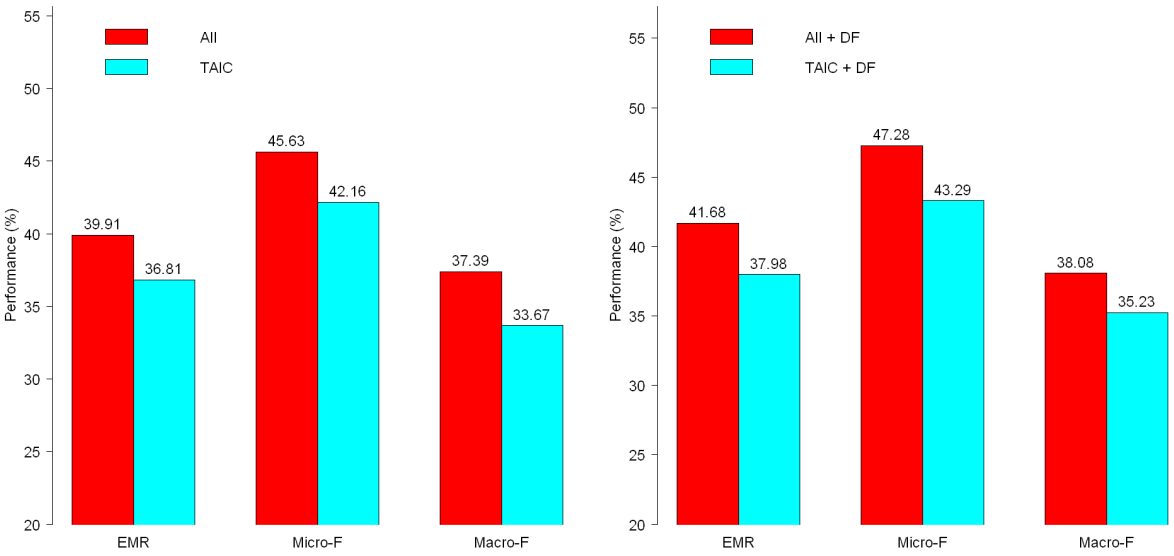

Figure 6.1: Effectiveness of Feature Selection based on Text Segmentation (ML-kNN, Binary Weighting)

However, in TF-IDF Weighting, TAIC model shows better results than All. Exact Match Ratio increases 4.2% and 1.02% by not using and using DF, respectively. Micro-F raises 4.97% and 2.15%. Macro-F increases 7.22% and 4.5%. Feature selection of TAIC model seems effective for TF-IDF Weighting, but not for Binary Weighting. Since results of Binary Weighting are better than those of TF-IDF Weighting, we can conclude that feature selection based on text segmentation is not so effective in ML-kNN.

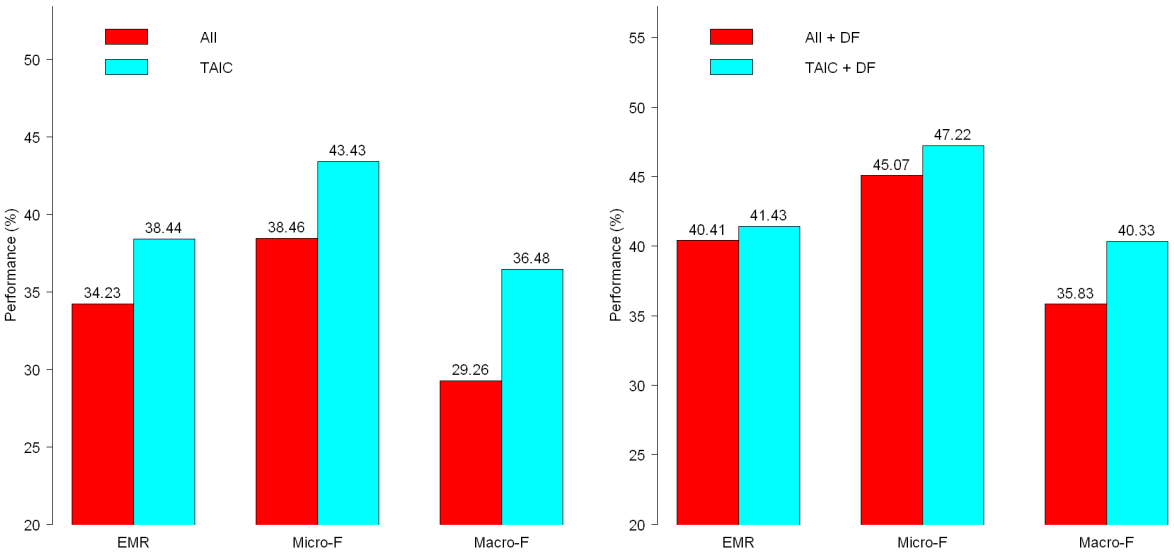

Figure 6.2: Effectiveness of Feature Selection based on Text Segmentation (ML-kNN, TF-IDF Weighting)

To assess the effectiveness of two features, Title Bi-Gram and Title SigNoun, we compare the results among TAIC, TAIC + Title SigNoun and TAIC + Title SigNoun, with and without dimentionality reduction by DF, in terms of three metrics (EMR, MicroF and MacroF). Figure 6.3 shows results for the above comparison, where TF-IDF Weighting is used as feature weighting.

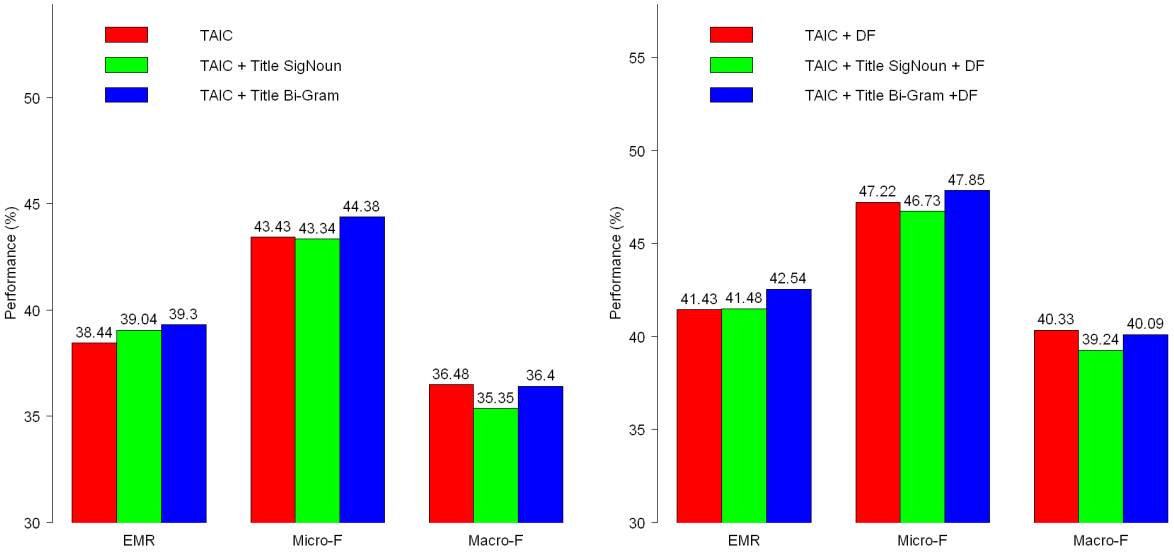

Figure 6.3: Effectiveness of Title Features (ML-kNN, TF-IDF Weighting)

Figure 6.3 indicates that there are a little improvement in Exact Match Ratio by Title Bi-Gram and Title SigNoun features. However, Title SigNoun decreased Micro-F and Macro-F, and Title Bi-Gram decreased Macro-F.

Binary Approach

Table 6.3 reveals results of Binary Approach. The values of the best system for each evaluation metrics is represented in bold.

Table 6.3: Results of Binary Approach

Term Selection	**TW**	**Metrics**									
		Instance-based				**Category-based**					
		EMR	**A**	**P**	**R**	**Mi-P**	**Mi-R**	**Mi-F**	**Ma-P**	**Ma-R**	**Ma-F**
All	BW	42.24	47.69	79.74	50.03	77.10	48.50	59.51	70.23	37.99	54.55
	TF-IDF	43.76	55.75	67.69	65.64	60.75	64.21	62.41	50.97	55.11	52.85
All + DF	BW	41.78	47.24	**79.78**	49.52	**77.21**	48.05	59.20	**70.58**	37.51	54.26
	TF-IDF	44.67	55.06	69.86	63.34	62.05	61.71	61.86	52.00	52.73	52.19
TAIC	BW	40.36	46.20	78.03	49.04	74.97	47.65	58.25	68.86	37.40	52.96
	TF-IDF	48.02	57.78	72.99	64.91	67.81	63.46	65.51	59.03	55.34	56.58
TAIC + DF	BW	39.86	45.74	78.25	48.61	75.01	47.26	57.97	68.67	37.20	52.69
	TF-IDF	46.14	55.18	74.00	61.30	68.71	59.91	63.95	61.48	51.71	56.43
TAIC + Title SigNoun	TF-IDF	49.24	59.00	74.17	66.03	69.12	64.65	66.77	60.50	55.73	58.08
TAIC + Title SigNoun + DF	TF-IDF	47.72	56.63	75.16	62.75	69.97	61.55	65.44	61.86	53.35	57.44
TAIC + Title Bi-Gram	TF-IDF	**49.34**	**59.43**	75.01	**66.63**	69.78	**65.14**	**67.34**	60.91	**57.25**	58.46
TAIC + Title Bi-Gram + DF	TF-IDF	48.94	57.80	75.71	63.84	70.73	62.26	66.19	63.51	53.06	**58.95**

Figure 6.4 and 6.5 compare results of All and TAIC or All + DF and TAIC + DF, using Binary and TFIDF feature weighting, to evaluate effectiveness of feature selection

based on text segmentation. Similar to ML-kNN method, in Binary Weighting method, All beats TAIC method by 1.88% (without DF) and 1.92% (with DF) in Exact Match Ratio, 1.26% and 1.25% in Micro-F, 1.59% and 1.57% in Macro-F.

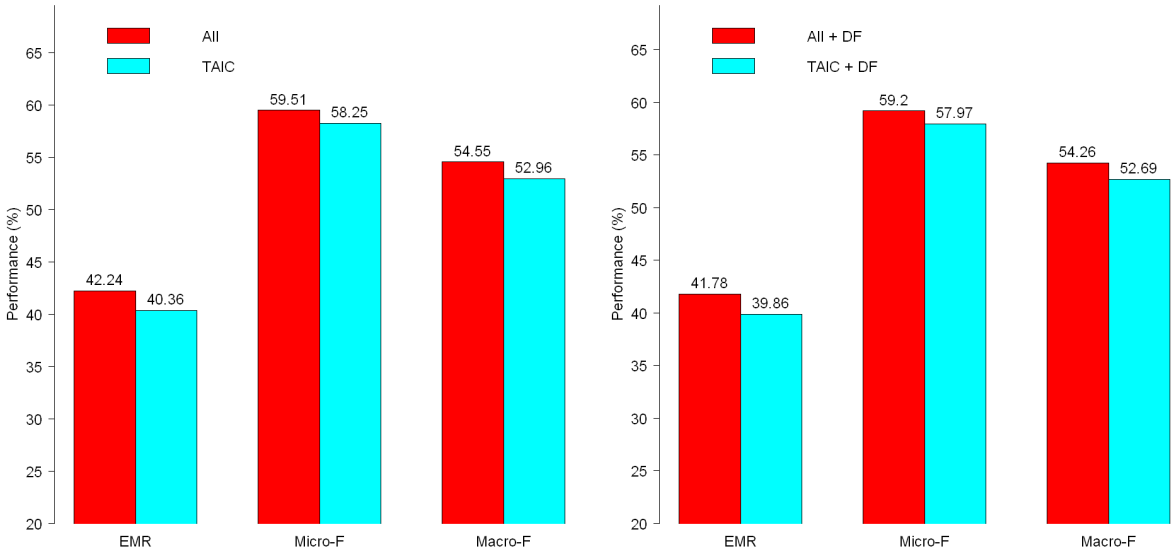

Figure 6.4: Effectiveness of Feature Selection based on Text Segmentation (Binary Approach, Binary Weighting)

In contrast, in TF-IDF Weighting, TAIC gives better results than All. Exact Match Ratio rises 4.26% and 1.47% by not using and using DF, respectively. Micro-F increases 3.1% and 2.09%. Macro-F increases 3.73% and 4.24%.

In both ML-kNN and Binary Approach, feature selection based on text segmentation is effective in TF-IDF Weighting, but not in Binary Weighting. In addition, unlike ML-kNN, the results in TF-IDF are more likely to be better than those in Binary Weighting. Furthermore, when we compare ML-kNN with Binary Weighting and Binary Approach with TF-IDF Weighting, the latter is better. Even though TAIC model is not effective in Binary Weighting, we can conclude that it is effective in our models.

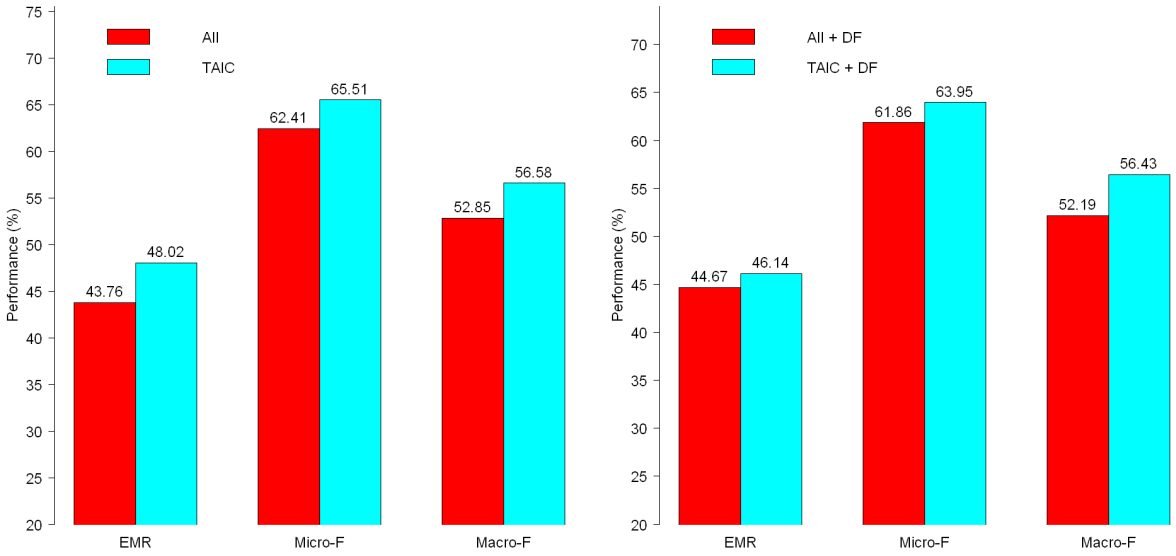

Figure 6.5: Effectiveness of Feature Selection based on Text Segmentation (Binary Approach, TF-IDF Weighting)

Figure 6.6 compares TAIC, TAIC + Title SigNoun and TAIC + Title Bi-Gram, with and without dimentionality reduction by DF, to evaluate the contribution of title features.

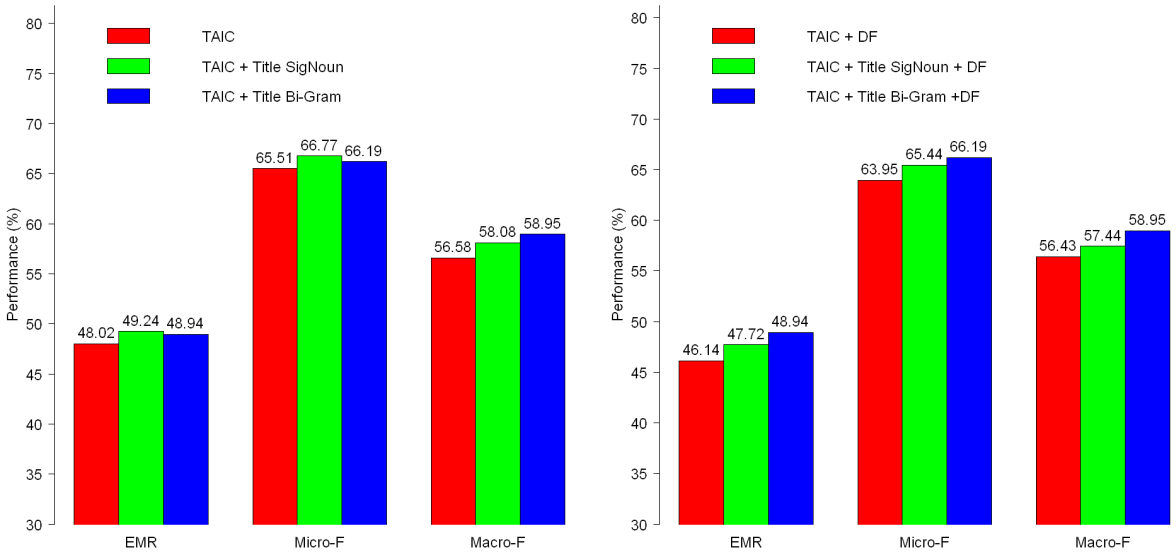

Figure 6.6: Effectiveness of Title Features (Binary Approach, TF-IDF Weighting)

It indicates that combining two features Title Bi-Gram and Title SigNoun improves the performance on three metrics. Therefore, we can conclude that our new features derived from the title are effective in this model. Comparing results in Table 6.2 and 6.3, Exact Match Ratio (EMR) and Micro-F of Binary Approach of TAIC + Title Bi-Gram model were 6.8% and 19.49% better than ML-kNN method. In both models, the feature sets with Title Bi-Gram are more likely to give better results than others.

Back-off Model

Back-off Model is built based on Binary Approach. Moreover, from Table 6.3 in Binary Approach, the better models are ones using words only in Title, Abstract, Introduction and Conclusion as feature selection, using Title SigNoun or Title Bi-Gram, using TF-IDF as feature weighting and not using dimensionality reduction by DF. Therefore, in the Back-off Model, we choose these models to conduct the experiment.

Table 6.4 shows results of Back-off Model. The values of the best system for each evaluation metrics is represented in bold. Model 1, 2, 3 and 4 mean Back-off models explained in Subsection 6.1.1. Threshold T_1, T_2, T_3, T_4 ($100 \geq T_1 \geq T_2 \geq T_3 \geq T_4 \geq 50$) are choosen based on our intuition .

Table 6.4: Results of Back-off Model

Model	**Thresholds** T_1-T_2-T_3-T_4	**Metrics**									
		Instance-based				**Category-based**					
		EMR	**A**	**P**	**R**	**Mi-P**	**Mi-R**	**Mi-F**	**Ma-P**	**Ma-R**	**Ma-F**
1	80-80-50-50	57.10	63.99	75.63	66.70	72.89	64.56	68.45	66.38	55.74	61.12
	80-80-70-50	56.64	63.14	76.40	65.26	74.51	63.19	68.37	67.99	54.26	61.27
	80-80-80-50	56.59	62.95	76.52	64.98	74.70	62.84	68.23	68.15	53.99	61.22
2	80-80-50-50	57.45	65.07	70.07	**68.02**	67.50	**65.57**	66.51	61.22	**57.06**	59.23
	80-80-70-50	**58.31**	**65.24**	70.29	67.34	68.57	64.91	66.68	62.24	56.62	59.80
	80-80-80-50	58.21	65.12	70.18	67.29	68.21	64.87	66.49	62.06	56.61	59.69
3	80-80-50-50	58.21	65.10	77.34	67.57	75.03	65.27	**69.78**	69.27	57.04	62.56
	80-80-70-50	58.01	64.51	77.80	66.50	76.03	64.21	69.58	70.33	55.99	62.43
	80-80-80-50	58.16	64.49	78.07	66.37	76.42	64.12	69.68	**71.01**	55.92	**62.68**
4	80-80-50-50	56.29	63.01	77.14	65.29	74.25	63.04	68.16	68.38	54.14	61.52
	80-80-70-50	56.74	62.63	**78.98**	64.08	76.97	61.80	68.53	70.63	53.36	62.08
	80-80-80-50	56.59	62.48	78.94	63.96	**77.03**	61.63	68.45	70.93	53.24	62.20

Seeing results in Table 6.3 and 6.4, Exact Match Ratio (EMR), Micro-F and Macro-F of Back-off model were increased by 8.97%, 2.44% and 3.73% compared with Binary Approach, respectively. Therefore, we conclude that Back-off model is better than Binary Approach.

To compare the performance of three models (ML-kNN, Binary Approach and Back-off model) in detail, we plot the highest performance of all metrics for each method in Figure 6.7. It indicates that ML-kNN performs much worse than Binary Approach and Back-off Model on all metrics. Binary Approach method beats Back-off Model on Precision and Micro-Precision metrics. In contrast, Back-off Model tends to achieve better results on

Exact Match Ratio, Accuracy, Recall, Micro-Recall, Micro-F and Macro-F. Therefore, Back-off Model is the best model among three approaches.

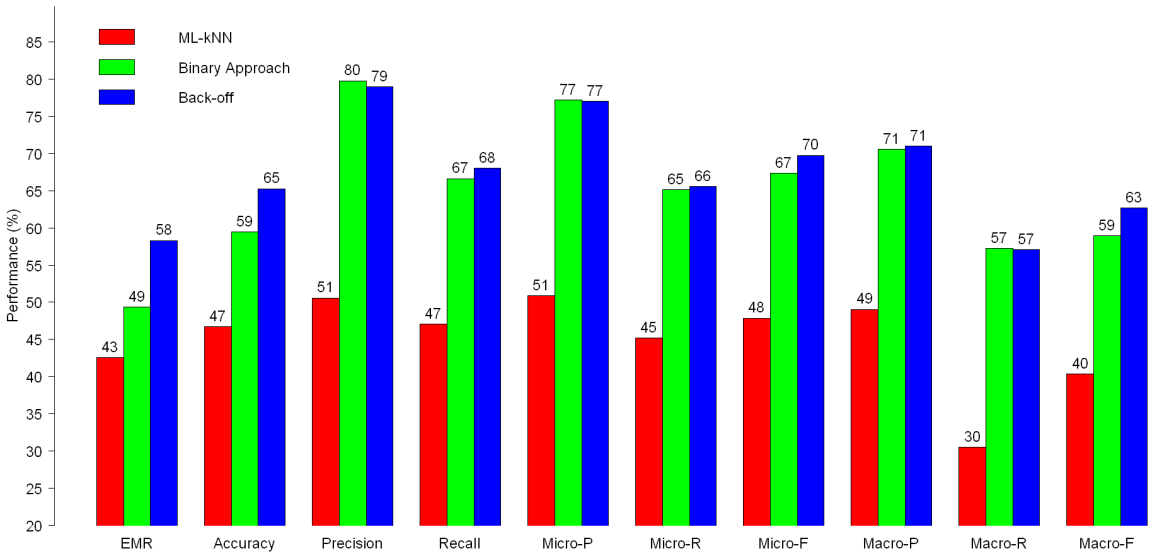

Figure 6.7: Best Performance of Three Models

6.2 Evaluation of Subtopic Key Phrase Extraction

6.2.1 Experiment Setup

As pointed out in Section 4.1, papers have well-organized structure and research topic is often described in the title, abstract and so on. Therefore, we use words only in Title, Abstract, Introduction and Conclusion sections for TextRank and SingleRank. The vertices added to the graph are restricted with a filter, which selects only lexical units of a certain part of speech. In these two models, only adjectives or nouns are selected as vertices.

TextRank

The first parameter in TextRank is window size. It refers to the size of the co-occurence window. After trying values N = 2,3,5 and 10 as the window size, we found that window size N = 3 seems the best. Therefore, we choose window size N = 3 for TextRank model.

For TextRank, instead of the number predicted key phrases, the percentage of top scored vertices that are selected as keywords is used for the second parameter. We set this parameter to 5%.

SingleRank

Same as in TextRank model, we also choose the window size = 3 for SingRank model.

Another parameter is the number of extracted key phrases per a paper. This parameter denotes the number of key phrases to be extracted from each input file. To prevent subtopics from not being extracted, we set it to 10. The extracted key phrases may be overlapped, however, the postprocessing will remove such key phrases.

Metrics

Evaluation of key phrase extraction is subjective and difficult. The most difficulty for evaluating the performance of subtopic key phrase extraction is that we don't have the gold key phrases which are considered correct subtopics in each paper. Constructing such gold key phrases is time-consuming and not easy at all. In this experiment, we used a subset of full dataset consisting of 50 papers from ACL 2010 conference for calculating precision. Precision is defined as:

$$\textbf{Precision} = \frac{1}{N}\sum_{i=1}^{N} P_i \tag{6.11}$$

where N is the total number of papers, and P_i is precision of extracted key phrases from paper i, defined as Equation (6.12).

$$P_i = \frac{\#\ of\ correct\ key\ phrases}{\#\ of\ key\ phrases} \tag{6.12}$$

We manually judge whether each output of key phrase is correct or not to calculate P_i in Equation (6.12).

6.2.2 Results

Table 6.5 reveals Precision of TextRank and SingleRank as well as the number of extracted key phrases and key phrases judged as correct by human in 50 papers.

Table 6.5: Evaluation of Subtopic Key Phrase Extraction

Model	# of Key Phrases	# of Accepted Key Phrases	Precision
TextRank	104	67	66%
SingleRank	81	65	79%

From Table 6.5, SingleRank model achieved better Precision than TextRank model. However, with only Precision metric, it is rather hard to say which model is better. Because we don't have the gold key phrases, we cannot evaluate Recall metric for these models. It will be future work of this thesis.

We show some examples of extracted subtopics in two models:

1.
- ID: P10-1040
- Title: Word Representations: A Simple and General Method for Semi-Supervised Learning
- Categories: Morphology
- Extracted Subtopics: Table 6.6

Table 6.6: Extracted Subtopics of the Paper P10-1040

Model	Extracted SubTopic	Accepted
TextRank	unsupervised word representation	1
	word feature	0
SingleRank	unsupervised word representation	1
	certain word feature	0

2.
- ID: P10-1034
- Title: Fine-Grained Tree-to-String Translation Rule Extraction
- Categories: Machine Translation, SpeechToSpeech Translation
- Extracted Subtopics: Table 6.7

Table 6.7: Extracted Subtopics of the Paper P10-1034

Model	Extracted SubTopic	Accepted
TextRank	fine-grained tree-to-string translation rule	1
	fine-grained translation rule	1
SingleRank	fine-grained tree-to-string translation rule extraction	1
	fine-grained translation rule set	1
	syntax-based statistical machine translation system	1

3.
- ID: P10-1004
- Title: Computing Weakest Readings
- Categories: Semantics
- Extracted Subtopics: Table 6.8

Table 6.8: Extracted Subtopics of the Paper P10-1004

Model	Extracted SubTopic	Accepted
TextRank	semantic representation	1
SingleRank	semantic representation	1

Chapter 7

Conclusion

7.1 Contribution

Automatically understanding research topics in papers is challenging and a difficult problem. This thesis tries to solve this problem through two tasks: multi-label classification and subtopic key phrase extraction. We summarize our contributions as follows:

1. Constructed the multi-label data by collecting technical papers in proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL) from 2000-2011. We manually assigned the research topics for each paper in this collection.

2. Evaluated the effectiveness of text segmentation and text representations with three different models in multi-label classification. New features Title SigNoun and Title Bi-Gram are used to improve the performance. The results of the experiments show that feature selection based on text segmentation and these two features are effective in our models. In addition, we proposed a novel method for text classification based on the structure of papers. The results show that this model outperforms than ML-kNN and Binary Approach.

3. Extracted the subtopics of papers through key phrase extraction algorithms. From the view point of broad range of categories in multi-label classification, this task supports the first task by providing us specific subtopics of papers.

7.2 Future Work

There are several drawbacks in the proposed method. In this thesis, we have mainly investigated in the first task: multi-label classification of technical papers. The second task, subtopic key phrase extraction, still has many weaknesses in evaluation step. The next study in this research will focus on the following issues:

- Design an effective method of feature selection and feature weighting to improve the accuracy of text classification. For example, combining Topic Modeling such as Latent Dirichlet Allocation [21] to exploit the semantics of content will be considered.

- Order the categories according to their relevance to the paper.
- Construct gold key phrases to evaluate the performance of subtopic key phrase extraction of technical papers more precisely.
- Propose a novel method for subtopic key phrases extraction of technical papers.

Bibliography

[1] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, pp. 1–47, Mar. 2002.

[2] A. Rahmoun and Z. Elberrichi, "Experimenting n-grams in text categorization," *Int. Arab J. Inf. Technol.*, pp. 377–385, 2007.

[3] M. D. Cao and X. Gao, "Combining contents and citations for scientific document classification.," in *Australian Conference on Artificial Intelligence'05*, pp. 143–152, 2005.

[4] M. Zhang, X. Gao, M. D. Cao, and Y. Ma, "Modelling citation networks for improving scientific paper classification performance," in *Proceedings of the 9th Pacific Rim international conference on Artificial intelligence*, PRICAI'06, (Berlin, Heidelberg), pp. 413–422, Springer-Verlag, 2006.

[5] N. T. and M. Y., "Exploiting text structure for topic identification," in *Proceedings of the 4th Workshop on Very Large Corpora*, pp. 101–112, 1996.

[6] L. S. Larkey, "A patent search and classification system," in *Proceedings of the fourth ACM conference on Digital libraries*, DL '99, (New York, NY, USA), pp. 179–187, ACM, 1999.

[7] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," in *Data Mining and Knowledge Discovery Handbook* (O. Maimon and L. Rokach, eds.), pp. 667–685, Springer US, 2010.

[8] R. E. Schapire and Y. Singer, "Boostexter: A boosting-based system for text categorization," in *Machine Learning*, pp. 135–168, 2000.

[9] E. Spyromitros, G. Tsoumakas, and I. Vlahavas, "An empirical study of lazy multilabel classification algorithms," in *Proceedings of the 5th Hellenic conference on Artificial Intelligence: Theories, Models and Applications*, SETN '08, (Berlin, Heidelberg), pp. 401–406, Springer-Verlag, 2008.

[10] M.-L. Zhang and Z.-H. Zhou, "Ml-knn: A lazy learning approach to multi-label learning," *Pattern Recognition*, vol. 40, no. 7, pp. 2038 – 2048, 2007.

[11] T. D. Nguyen and M. yen Kan, "Keyphrase extraction in scientific publications," in *In Proc. of International Conference on Asian Digital Libraries (ICADL 07*, pp. 317–326, Springer, 2007.

[12] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning, "Kea: practical automatic keyphrase extraction," in *Proceedings of the fourth ACM conference on Digital libraries*, DL '99, (New York, NY, USA), pp. 254–255, ACM, 1999.

[13] P. D. Turney, "Learning to Extract Keyphrases from Text," tech. rep., National Research Council, Institute for Information Technology, Dec. 1999.

[14] J. Park, J. G. Lee, and B. Daille, "Unpmc: Naive approach to extract keyphrases from scientific articles," in *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, (Stroudsburg, PA, USA), pp. 178–181, Association for Computational Linguistics, 2010.

[15] Y. Matsuo and M. Ishizuka, "Keyword extraction from a single document using word co-occurrence statistical information," *International Journal on Artificial Intelligence Tools*, vol. 13, p. 2004, 2004.

[16] K. S. Hasan and V. Ng, "Conundrums in unsupervised keyphrase extraction: Making sense of the state-of-the-art," in *Proceedings of COLING 2010: Posters Volume*, pp. 365–373, 2010.

[17] R. Mihalcea and P. Tarau, "Textrank: Bringing order into texts," in *Proceedings of EMNLP 2004* (D. Lin and D. Wu, eds.), (Barcelona, Spain), pp. 404–411, Association for Computational Linguistics, July 2004.

[18] X. Wan and J. Xiao, "Single document keyphrase extraction using neighborhood knowledge," in *Proceedings of the 23rd national conference on Artificial intelligence - Volume 2*, AAAI'08, pp. 855–860, AAAI Press, 2008.

[19] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas, "Mulan: A java library for multi-label learning," *Journal of Machine Learning Research*, vol. 12, pp. 2411–2414, 2011.

[20] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[21] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

Appendix A

The Mapping File between Session Titles and Categories

Table A.1: The Mapping File

Section	**Categories**
algorithms	
alignment for machine translation	[Machine Translation, SpeechToSpeech Translation]
applications	[Tools, systems, applications]
asian language processing	
asian languages	
best asian language paper nominees	
best paper session	
categorial grammar	[Grammar and Syntax]
chunk parsing	[Chunking]
chunking	[Chunking]
chunking and tagging	[Chunking]
conversational spoken language processing	[Dialogue]
	[Speech Recognition/Understanding]
coreference	[Anaphora, Coreference]
coreference and anaphora	[Anaphora, Coreference]
coreference in discourse and dialogue	[Anaphora, Coreference]
	[Dialogue]
coreference resolution	[Anaphora, Coreference]
corpora	[Corpus (creation, annotation, etc.)]
corpus & document analysis	[Corpus (creation, annotation, etc.)]
corpus annotation	[Corpus (creation, annotation, etc.)]
data oriented parsing	[Parsing]
decipherment	

demo session	
demonstration	
demos	
dialog systems	[Dialogue]
dialogue	[Dialogue]
dialogue and generation	[Dialogue]
	[Natural Language Generation]
dialogue systems	[Dialogue]
disambiguation	
discourse	[Discourse annotation, representation and processing]
discourse & pragmatics	[Discourse annotation, representation and processing]
discourse and dialog	[Dialogue]
	[Discourse annotation, representation and processing]
discourse and dialogue	[Dialogue]
	[Discourse annotation, representation and processing]
discourse and dialogue segmentation	[Dialogue]
	[Discourse annotation, representation and processing]
error correction	[Error Correction]
error detection in spoken dialog systems	[Dialogue]
evaluation	[Evaluation methodologies]
evaluation of machine translation	[Evaluation methodologies]
evaluation systems	[Evaluation methodologies]
event-role extraction	[Information Extraction]
formal grammars	[Grammar and Syntax]
generation	[Natural Language Generation]
generation and summarization	[Natural Language Generation]
	[Summarisation]
generation and summarization	[Natural Language Generation]
	[Summarisation]
generation/paraphrasing	[Natural Language Generation]
	[Textual Entailment and Paraphrasing]
grammar	[Grammar and Syntax]
grammar and the lexicon	[Grammar and Syntax]
	[Lexicon, lexical database]
grammar formalisms	[Grammar and Syntax]
grammars	[Grammar and Syntax]
historical linguistics	

information extraction	[Information Extraction]
information extraction/information retrieval	[Information Extraction]
	[Information Retrieval]
information retrieval	[Information Retrieval]
information retrieval and extraction	[Information Extraction]
	[Information Retrieval]
information retrieval systems	[Information Retrieval]
information retrieval/information extraction	[Information Extraction]
	[Information Retrieval]
interactive poster	
interactive poster / demo	
invited talk	
knowledge base extension	
language acquisition	[Acquisition]
language generation	[Natural Language Generation]
language learning and models of language	[Language modelling]
language modeling	[Language modelling]
language modelling	[Language modelling]
language models	[Language modelling]
language resource	[LR Infrastructures and Architectures]
	[LR national/international projects, organizational/policy issues]
	[Standards for LRs]
language resources	[LR Infrastructures and Architectures]
	[LR national/international projects, organizational/policy issues]
	[Standards for LRs]
language resources and evaluation	[Evaluation methodologies]
	[LR Infrastructures and Architectures]
	[LR national/international projects, organizational/policy issues]
	[Standards for LRs]
lexica and ontologies	[Lexicon, lexical database]
	[Ontologies]
lexical acquisition from corpora	[Acquisition]
lexical issues	[Lexicon, lexical database]
lexical semantics	[Lexicon, lexical database]
lexical semantics and similarity	[Lexicon, lexical database]
lexicon	[Lexicon, lexical database]
lexicon and lexical semantics	[Lexicon, lexical database]
linguistic and mathematical models of language	[Language modelling]

linguistic creativity	
linguistic kinships	
machine learning	[Statistical and machine learning methods]
machine learning and statistical methods	[Statistical and machine learning methods]
machine learning in dialogue	[Dialogue]
	[Statistical and machine learning methods]
machine learning methods	[Statistical and machine learning methods]
machine learning, corpus and information retrieval	[Corpus (creation, annotation, etc.)]
	[Information Extraction]
	[Statistical and machine learning methods]
machine learning: kernels	[Statistical and machine learning methods]
machine translation	[Machine Translation, SpeechToSpeech Translation]
machine translation and chunking	[Machine Translation, SpeechToSpeech Translation]
	[Chunking]
machine translation and multilinguality	[Machine Translation, SpeechToSpeech Translation]
	[Multilinguality]
machine translation i	[Machine Translation, SpeechToSpeech Translation]
machine translation systems	[Machine Translation, SpeechToSpeech Translation]
machine translation	[Machine Translation, SpeechToSpeech Translation]
morphology	[Morphology]
morphology and word segmentation	[Morphology]
	[Segmentation]
morphology/pos induction	[Morphology]
mt: methods	[Machine Translation, SpeechToSpeech Translation]
mt: models & evaluation	[Machine Translation, SpeechToSpeech Translation]
	[Evaluation methodologies]
mt: reordering models	[Machine Translation, SpeechToSpeech Translation]
multi-modality	[Multimodal]
multilingual lexicons	[Lexicon, lexical database]
multilinguality	[Multilinguality]
multimodal	[Multimodal]
multimodal and situated language processing	[Multimodal]
multimodal systems	[Multimodal]

multimodality	[Multimodal]
named entities and bootstraping	[Named Entity recognition]
named entity and information extraction	[Information Extraction]
	[Named Entity recognition]
named entity detection	[Named Entity recognition]
natural language processing applications	[Tools, systems, applications]
nlp applications	[Tools, systems, applications]
nlp for web 2.0	[Tools, systems, applications]
nlp tools	[Tools, systems, applications]
ontologies	[Ontologies]
opinion analysis and textual	[Dialogue]
and spoken conversations	[Opinion mining / sentiment analysis]
paraphrasing, textual entailment	[Textual Entailment and Paraphrasing]
parsing	[Parsing]
parsing and generation	[Grammar and Syntax]
with lexicalized grammars	[Natural Language Generation]
	[Parsing]
parsing and grammar formalisms	[Grammar and Syntax]
	[Parsing]
parsing and grammars	[Grammar and Syntax]
	[Parsing]
parsing and semantics	[Parsing]
	[Semantics]
parsing and tagging	[Parsing]
	[Part of speech tagging]
parsing german	[Parsing]
parsing i	[Parsing]
part of speech tagging	[Part of speech tagging]
and spelling correction	[Error Correction]
part-of-speech tagging	[Part of speech tagging]
partial parsing	[Chunking]
phonetics	[Phonetic Databases, Phonology]
phonology	[Phonetic Databases, Phonology]
phonology and morphology	[Morphology]
	[Phonetic Databases, Phonology]
phonology, morphology	[Morphology]
	[Phonetic Databases, Phonology]
phonology, word segmentation	[Part of speech tagging]
and pos tagging	[Phonetic Databases, Phonology]
	[Segmentation]
phonology/morphology & postagging	[Morphology]
	[Part of speech tagging]

	[Phonetic Databases, Phonology]
poster and demo session	
poster presentations	
poster session student research workshop	
poster_session	
presentation of awards	
probabilistic parsing	[Parsing]
psycholinguistics	[Other]
quantitative and formal linguistics	[Other]
question answering	[Question Answering]
question answering and entailment	[Question Answering]
	[Textual Entailment and Paraphrasing]
question answering systems	[Question Answering]
question-answering	[Question Answering]
relation extraction	[Information Extraction]
resource development systems	
resources	
resources and evaluation	
resources and mt evaluation	
rule-based parsing	[Parsing]
segmentation, tagging, and semantic role labeling	[Segmentation]
	[Semantic role labeling]
segments and segmentation	[Segmentation]
selectional preferences	
semantic relations	[Semantics]
semantic role labeling	[Semantic role labeling]
semantics	[Semantics]
semantics i	[Semantics]
sentiment	[Opinion mining / sentiment analysis]
sentiment analysis	[Opinion mining / sentiment analysis]
sentiment analysis and text categorization	[Document Classification, Text categorisation]
	[Opinion mining / sentiment analysis]
sentiment analysis/opinion mining	[Opinion mining / sentiment analysis]
sequence processing	
session 7d:short paper 11 (information extraction)	[Information Extraction]
short papers 1 (machine translation)	[Machine Translation, SpeechToSpeech Translation]
short papers 1 (syntax)	[Grammar and Syntax]
short papers 2 (dialog/statistical methods)	[Dialogue]
	[Statistical and machine learning methods]

short papers 2 (speech)	[Speech Recognition/Understanding]
	[Speech resource/database]
	[Speech Synthesis]
short papers 3 (semantics)	[Semantics]
short papers 3 (semantics/phonology)	[Phonetic Databases, Phonology]
	[Semantics]
short papers 4 (generation/summarization)	[Natural Language Generation]
	[Summarisation]
short papers 4 (ir/sentiment analysis)	[Information Extraction]
	[Opinion mining / sentiment analysis]
smoothing	[Language modelling]
smt: phrase-based models	[Machine Translation, SpeechToSpeech Translation]
smt: tree-based models	[Machine Translation, SpeechToSpeech Translation]
software demonstration session	
speech	[Speech Recognition/Understanding]
	[Speech resource/database]
	[Speech Synthesis]
speech and language modeling	[Language modelling]
	[Speech Recognition/Understanding]
	[Speech resource/database]
	[Speech Synthesis]
speech and multimodal	[Speech Recognition/Understanding]
	[Speech resource/database]
	[Speech Synthesis]
	[Multimodal]
speech dialogue	[Dialogue]
	[Speech Recognition/Understanding]
	[Speech resource/database]
	[Speech Synthesis]
speech processing	[Speech Recognition/Understanding]
	[Speech resource/database]
	[Speech Synthesis]
spoken dialog	[Dialogue]
	[Speech Recognition/Understanding]
	[Speech resource/database]
	[Speech Synthesis]
spoken language	[Speech Recognition/Understanding]
	[Speech resource/database]
	[Speech Synthesis]
spoken language processing	[Speech Recognition/Understanding]

	[Speech resource/database]
	[Speech Synthesis]
srw 1: multilinguality	[Multilinguality]
srw 2: speech	[Speech Recognition/Understanding]
	[Speech resource/database]
	[Speech Synthesis]
srw 3: parsing	[Parsing]
statistical and machine learning methods	[Statistical and machine learning methods]
statistical machine translation	[Machine Translation, SpeechToSpeech Translation]
statistical modeling	[Statistical and machine learning methods]
statistical parsing	[Parsing]
statitical and machine learning methods	[Statistical and machine learning methods]
student research workshop	
student research workshop poster session	
student research workshop session 1	
student research workshop session 2	
student research workshop session 3	
student research workshop session a	
student research workshop session b	
student research workshop session c	
subcategorization and word meaning	[Lexicon, lexical database]
summarization	[Summarisation]
summarization & generation	[Natural Language Generation]
	[Summarisation]
summarization and generation	[Natural Language Generation]
	[Summarisation]
syntax	[Grammar and Syntax]
syntax & parsing	[Grammar and Syntax]
	[Parsing]
syntax & parsing	[Grammar and Syntax]
	[Parsing]
syntax and parsing	[Grammar and Syntax]
	[Parsing]
syntax/semantics/parsing	[Grammar and Syntax]
	[Parsing]
	[Semantics]
syntax/semantics/parsing (grammar construction)	[Grammar and Syntax]
	[Parsing]
	[Semantics]
tagging	[Part of speech tagging]
tagging and chunking	[Chunking]

techniques and systems	[Tools, systems, applications]
techniques and tools	[Tools, systems, applications]
term generation	[Natural Language Generation]
text categorization	[Document Classification, Text categorisation]
text classification	[Document Classification, Text categorisation]
text classication and topic models	[Document Classification, Text categorisation]
text mining and nlp applications	[Text mining]
	[Tools, systems, applications]
text mining and retrieval	[Information Extraction]
	[Text mining]
textual entailment	[Textual Entailment and Paraphrasing]
tools and systems	[Tools, systems, applications]
topic segmentation	[Topic detection & tracking]
topic spotting and language acquisition	[Acquisition]
	[Document Classification, Text categorisation]
translation	[Machine Translation, SpeechToSpeech Translation]
translation and multilinguality	[Machine Translation, SpeechToSpeech Translation]
	[Multilinguality]
transliteration/alignment	[Machine Translation, SpeechToSpeech Translation]
tree transducers	
unsupervised parsing and grammar induction	[Grammar and Syntax]
	[Parsing]
vector space models	[Information Retrieval]
word segmentation	[Segmentation]
word segmentation and pos tagging	[Part of speech tagging]
	[Segmentation]
word segmentation for arabic	[Segmentation]
word sense disambiguation	[Word Sense Disambiguation]
word sense disambiguation and machine translation	[Machine Translation, SpeechToSpeech Translation]
	[Word Sense Disambiguation]