Multi-Dimensional Fragment Classification in Biomedical Text

By

Fengxia Pan

A thesis submitted to the

School of Computing

in conformity with the requirements for

the degree of Master of Science

Queen's University

Kingston, Ontario, Canada

September 2006

Abstract

Automated text categorization is the task of automatically assigning input text to a set of categories. With the increasing availability of large collections of scientific literature, text categorization plays a critical role in managing information and knowledge, and biomedical text categorization is becoming an important area of research. The work presented here is motivated by the possibility of using automated text categorization to identify and characterize information-bearing text within biomedical literature. Under a recently suggested classification scheme [ShWR06], we examine the feasibility of using machine learning methods to automatically classify biomedical sentence fragments into a set of categories, which were defined to characterize and accommodate certain types of information needs. The categories are grouped into five dimensions: *Focus*, *Polarity*, *Certainty*, *Evidence*, and *Trend*. We conduct experiments using a set of manually annotated sentences that were sampled from different sections of biomedical journal articles. A classification model based on Maximum Entropy, designed specifically for this purpose, as well as two other popular algorithms in the area of text categorization, Naïve Bayes and Support Vector Machine (SVM), are trained and evaluated on the manually annotated dataset. The preliminary results show that machine learning methods can classify biomedical text along certain dimensions with good accuracy.

Acknowledgements

I would like to express my sincerest thanks to my supervisor, Hagit Shatkay, for her great guidance and endless support throughout the course of this thesis work. I would like to thank her for all her help and advice over the years I have pursued my education and research at Queen’s University.

I would like to thank my thesis examination committee, Dr. Skillicorn, Dr. Lessard, Dr. Tennent, and Dr. Singh, for their insightful suggestions.

I gratefully thank my friends Jess Shen, Henry Xiao, and Matt Kirkey for all of the helpful comments.

I would also like to extend my gratitude to the School of Computing at Queen’s University for funding my studies.

Finally, I would like to express my sincerest appreciation and love to my husband, Fei Zong, for his unwavering support and love.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Automated text categorization can be defined as the task of automatically assigning input text to a set of categories. The categories can be either pre-defined, a task usually called *text classification*, or automatically identified, a task called *text clustering*. With the increasing availability of large collections of scientific literature, automated text categorization plays a critical role in managing information and knowledge, and therefore becomes an important area of research. The work presented here focuses on the categorization of biomedical text.

1.1 Background

Text categorization dates back to the early 60's. At an early stage, the most popular approach to categorize text was to manually build a classifier (e.g. an expert system) consisting of a set of pre-defined rules. Building classifiers by hand requires domain knowledge and is time consuming. Moreover, when the categories or the nature of the data changes over time, it is difficult to update existing classifiers. Therefore, it is desirable to build and update a classifier automatically from existing data examples.

Since the early 90's, the machine learning approach has gradually gained popularity. Machine learning is concerned with constructing computer programs that can adapt and learn from past experience to solve a given problem. The programs are usually based on a learning algorithm. Using machine learning terminology, the process that deals with *classification* is called *supervised learning*, whereas the process that deals with *clustering* is called *unsupervised learning*.

Most work on text categorization focuses on *supervised learning*. Within this framework, a set of data examples are first manually classified and labeled with predefined categories by human experts. A learning algorithm is then applied to learn the characteristics of each category, and finally a classification model (classifier) is automatically built to decide the categories of future unknown data. Usually the sample dataset is divided into two parts, a training set, which is used to build classifiers by learning the characteristics of the categories, and a test set, which is used to test the performance of the classifiers. Nowadays, classifiers automatically built using machine learning techniques achieve high level of effectiveness and are dominant in the area of text categorization [Yang97, Seba99].

Text categorization has been applied to many tasks such as document indexing [YaCh92], document filtering [TRFT02], and database annotation [TRGN05]. This study addresses the possibility of using text categorization to identify information-bearing text within biomedical literature. Specifically, we examine the feasibility of using machine learning methods to automatically classify sentence fragments into a set of categories that are defined to characterize certain types of information needs.

1.2 Motivation

Our work is motivated by the need to locate important scientific facts within large amount of biomedical text. With the tremendous growth in the number of biomedical publications, it is becoming increasingly challenging to access valuable and reliable knowledge from an overwhelming range of text resources. Several recent evaluation efforts focusing on biomedical text mining [KDDC02, BioC04, TRGN05] suggest that there is much room and need for improvement. To identify and characterize text that satisfies certain types of information needs within biomedical literature, Wilbur *et al.* [WiRS06, ShWR06] propose a scheme to manually annotate text along five dimensions. These dimensions are defined as: the *Focus* of a fragment (*Scientific finding*, *General knowledge*, or *Methodology*), the *Polarity* (*Positive* vs. *Negative*) of a statement, the *Certainty* degree of an assertion (*Complete uncertainty*, *Low certainty*, *High likelihood*, and *Complete certainty*), the availability of supporting *Evidence* (*No evidence*, *Implication of evidence*, *Explicit citation*, and *Explicit evidence*), and the future *Trend* (*Increase* or *Decrease*) of a certain phenomenon. Since the value along each dimension can change mid-sentence, the basic annotation unit is defined as a sentence fragment. The user can customize his/her own scoring scheme to highlight categories of interest. Based on the annotation, the importance of a fragment can be evaluated and informative fragments that best satisfy the user's needs can be identified.

Our general goal is to automatically identify information-bearing sentence fragments within scientific text, that is, automatically annotate sentence fragments. The task of automatic fragment annotation can be divided into three subtasks. First, a sizable training corpus is manually annotated. Second, automated text classifiers are trained on the annotated data to classify fragments along the five dimensions defined above. Third, raw text files are automatically fragmented and annotated with the trained classifiers. This study addresses the second subtask.

The goal of this thesis is to examine the feasibility and reliability of automatic fragment annotation, that is, using machine learning methods to automatically categorize sentence fragments along multiple dimensions. We refer to this work as multi-dimensional fragment classification. Multi-dimensional fragment classification characterizes each fragment from various perspectives and enables the substantiation of knowledge at the sentential level, which is likely to serve a variety of applications and lead to more accurate extraction or retrieval of information from text. The classification scheme [WiRS06, ShWR06] is originally designed for biomedical literature, but it can be extended to other domains.

1.3 Related Work

Many existing methods, although different from our intent, can be used to look for specific information within a document, for example, named entity and relation extraction [Leek97, CoNT00, RaCr01], rhetorical analysis [TeMo99, MaEc02, McSr03, MiCo04b], and text summarization [Luhn58, Edmu68, KuPC95]. However, these approaches have some limitations.

The tasks of named entity and relation extraction aim to find structured data from unstructured text, for example, the identification of text related to genes, proteins, cells, as well as their interactions from biomedical literature. As the methods used in such tasks are tailored to specific needs, they typically involve specific terminology and accordingly their applicability is limited.

Rhetorical analysis of text units, which differentiates among rhetorical relations such as *antithesis*, *cause*, *elaboration*, and recognizes rhetorical zones such as *background*, *related work*, *method*, and *result*, enables the characterization of the sentence topic and allows the selection of particular sentence types. However, the definition of rhetorical categories needs to be comprehensive enough to distinguish sentences of interest from others. With the expansion of the categories, the complexity of the rhetorical analysis increases, and the feasibility of automating it decreases.

Automatic text summarization is concerned with selecting important text from documents. It typically measures the importance of a sentence based on a set of features including the frequencies of thematic words, sentence length and location, and the presence or absence of cue phrases (e.g. *in conclusion*, *this article*) or title/header words. This approach has several drawbacks for identifying important sentences. First, the criteria for importance are not specific enough to characterize individual sentences. Consequently, some important sentences may not be identified. Second, the definition of importance cannot discriminate among different types of informative sentences. Third, the way of determining importance is fixed, and cannot be customized according to a user's specific requirements.

Compared with existing work, the method we are investigating aims to characterize various types of information needs without losing generality, feasibility, and flexibility. The notion of multi-dimensional classification allows a comprehensive description of an entity from multiple perspectives, while maintaining sufficient generality of the category definition along each dimension. Accordingly, a broad range of factual information can be characterized at a relatively low level of complexity.

1.4 Contribution

In this thesis, we make several contributions. We first provide a broad survey of text categorization, followed by a comprehensive review of previous research on sentence classification. Under a recently suggested annotation scheme [ShWR06], we investigate a novel approach to classify the same text entity from several relatively independent dimensions. To the best of our knowledge, this idea has not been tried before. We study several techniques to preprocess the unstructured, free form text along different dimensions to optimize classification performance. To address several special challenges in the fragment classification task, we design a classification model that can take into account correlation within categories, correlation between dimensions, and dependence among fragments. We also propose a performance measure that can be applied to general multi-label classification tasks. Our experiments on a new and quite extensive dataset suggest that machine learning methods can automatically perform fragment annotation along certain dimensions with good accuracy.

1.5 Thesis Organization

This thesis is organized as follows: An overall review of text categorization and related work on sentence classification is presented in Chapter 2. An introduction to multi-dimensional fragment classification and its potential applications follow in Chapter 3. Chapters 4-5 discuss in detail the choice of preprocessing procedures and classification algorithms for each dimension. Experimental results and analysis are provided in Chapter 6, followed by future extensions and conclusions in chapters 7-8.

Chapter 2

Text Categorization: Background

In this chapter, we first define text categorization and briefly introduce the type of tasks it covers. We then discuss in detail the procedures involved in automatic text categorization, including text representation, classifier construction, and performance evaluation. Finally, we survey the related work on sentence categorization.

2.1 Introduction to Text Categorization

Text categorization can be defined as the task of automatically assigning input text to a set of categories. The categories can be either pre-defined, a task usually called *text classification*, or automatically identified, a task called *text clustering*. More formally, let $C = \{c_1, ..., c_{|C|}\}$ be a set of categories, and $D = \{d_1, ..., d_{|D|}\}$ be a set of data examples. Text categorization can be defined as the task of finding a function, f, which assigns a Boolean value to each pair $(d_j, c_i) \in D \times C$, with the value *True* indicating that data example d_j should be classified under category c_i, and the value *False* indicating that d_j does not belong to category c_i. The function f is called the classifier.

Text categorization can be further divided into several subtypes based on the constraints enforced on the tasks, such as the number of categories assigned to a data example, the structure of the categories, or the basic unit of classification.

2.1.1 Single-label vs. Multi-label Text Categorization

The case in which exactly one category is assigned to the input text is called *single-label* text categorization, whereas the case in which multiple categories can be assigned to the same input text is called *multi-label* text categorization [Seba99]. A simple form of single-label categorization is *binary* categorization, where each input text is either assigned to one category or to its complement. The most general application of *binary* categorization is *text filtering*, differentiating relevant documents from irrelevant ones according to a given topic [KOSL02, LeCL04]. An example application of multi-label text categorization is assigning a set of Gene Ontology (GO) terms [GO00] to an input document [EGJR05, RiNS05]. If the categories are mutually independent, multi-label categorization can be transformed into binary categorization by conducting multiple one-vs-rest *binary* categorizations independently. However, when there are potential correlations among different categories, multi-label categorization algorithms must be specifically designed to better capture such statistical constraints in the data [ZJXG05].

2.1.2 Soft vs. Hard Text Categorization

Hard categorization means clearly assigning one (or several) categories to an input example, while soft categorization means ranking the input examples or the output categories by the order of relevance, instead of making explicit assignment decision [Seba99]. There are two types of ranking categorization: category-ranking and data-ranking. Category-ranking ranks all categories according to the estimated probability that an input example belongs to them, whereas data-ranking ranks all data examples based on their relevance to a certain category. Compared to hard categorization, soft categorization can intuitively deliver more information about the level of certainty in the category assignment. Moreover, the curator can screen through the ranked list to decide whether the returned categories for a given data item, or the returned data examples for a specific category, are of interest or not.

Category-ranking is widely used in automatic document indexing, in which a set of key words from a controlled vocabulary is assigned to an input document as index terms. Since the size of the vocabulary is typically large, for example, of order 10^4 [YaCh92, WiYa96], only a few top ranking categories should be selected for further processing. Data-ranking categorization is especially helpful in the applications of document filtering, since it allows more flexible retrieval of relevant documents by returning the user-specified number of top ranking documents.

2.1.3 Flat vs. Hierarchical Text Categorization

Depending on the nature of the categories, text categorization can be either flat or hierarchical. Flat categorization treats all categories independently without considering any structural relation between them, while hierarchical categorization takes into account the semantics of a class tree or directed-acyclic graph (DAG) [SuLN03]. In some cases, the set of categories, for example, Gene Ontology (GO), is a hierarchy by its nature. Therefore a hierarchical framework that explores the semantics of the class hierarchy may be able to yield better performance than flat approaches.

The classification performance of the flat and hierarchical approaches was compared in the task of applying text categorization to associate genes with GO codes [RCSA02, KiMF04, KiMF05]. Two hierarchical classification methods have been investigated in an experiment by Kiritchenko *et al.* [KiMF04, KiMF05], *global* and *local*. The main idea of the *global* approach is to transform the hierarchical classification task into a multi-label classification task by expanding the initial label set with the corresponding ancestor label set for each data example. Only one classifier is built to discriminate all categories. A post-processing step is applied to re-label the examples whose label assignment is inconsistent with the category hierarchy. In the *local* approach, separate classifiers are built for each hierarchical level, and the classification proceeds in a top-down fashion. At each level, the classifier selects one or several (for the multi-label case) of the most likely categories and then proceeds down to inspect only the children of the selected nodes. The experimental result shows that a hierarchical classifier that incorporates the relationship among categories outperforms the flat one.

Hierarchical text classification has been explored by a number of authors [KoSa97, MRMN98, SaKi98, LaFi99, DuCh00, WaZH01, ViGi02]. It is still a relatively new field which is being actively pursued.

2.1.4 Sentence Level vs. Document Level Text Categorization

In terms of the span of the text to be classified, text categorization can be divided into document level categorization, namely, categorization of full text or paragraphs, and sentence level categorization, that is, categorization of sentences or sentence segments.

Document level text categorization typically considers the whole document as the basic classification unit and takes into account only the general subject of a document. It has been used in applications such as document indexing [YaCh92], document filtering [TRFT02], and database annotation [KDDC02, BioC04, TRGN05]. Document indexing is defined as automatically assigning a set of key words from a controlled vocabulary, such as MeSH [MeSH06] or GO [GO00], to characterize the content of the input text [IbCT99, SmCl03]. Document filtering means automatically selecting, for a given topic, only the articles that are relevant to it [CoMS04, PNRH05]. Database annotation refers to assigning attributes, for instance, biological process, molecular function, or cellular component, to entities such as genes or proteins in the database. Example annotation tasks that use document level text categorization include assigning GO codes to genes [RCSA02, KiMF04, EGJR05, KiMF05, RiNS05], discovering relationships among genes [SEWB00, JJDv05], and inferring sub-cellular localizations for proteins [NaRo02, EsAg04, HBBD06]. Typically a set of documents associated with a gene or protein is first obtained and a vector of words or phrases is constructed to represent the gene or protein. The task of annotating genes or proteins is then transformed into that of classifying or clustering the associated text.

In contrast to document level text categorization, sentence level text categorization breaks the classification granularity down to the sentence or sub-sentence level, and typically concerns the knowledge within the span of a sentence. Sentence level text categorization has been applied to many tasks, such as named entity and relationship extraction [Leek97, BSAG98, SkCR03], automatic text summarization [KuPC95], and rhetorical analysis [TeMo97, TeMo99, MaEc02, MiCo04a, MiCo04b]. As all the latter work is closely related to this thesis, these applications will be discussed in more detail in Section 2.3.

2.2 Procedures in Text Categorization

Typically there are three steps in automatic text categorization: text representation, classifier construction, and performance evaluation. We next examine each step closely.

2.2.1 Text Representation

Free text information is unstructured data. Text files vary in length and use different sets of words. As such, they cannot be readily interpreted by common classification algorithms. Therefore, preprocessing procedures that map the free text into a structured representation are necessary before applying classification algorithms. The most common way to represent text is based on the *bag of words* approach. It maps an input text (e.g. a document) to a vector of term weights, where terms can be words or phrases. The preprocessing thus includes term definition, term weight calculation, and term selection or extraction.

Term Definition

There are several ways to define terms, which may consider the lexical, semantic, syntactical or statistical information of the text. Among all the approaches, the most popular one is to generate terms based on individual words, and represent the input text as a vector of single words. However, this approach completely ignores the structural relationships among words, such as the dependencies and relative positions. To address this problem, a number of attempts have been made to use phrases, rather than individual words as terms. Phrases can be defined as either syntactical phrases or statistical phrases.

Syntactical phrases take into account the syntactical constraints among words, while statistical phrases capture their statistical co-occurrences.

A syntactical phrase is defined as a sequence of syntactically connected words, such as a *verb* phrase or a *noun* phrase. Lewis [Lewi92] studied the effect of using syntactical phrases instead of individual words as terms to represent the input text. The comparison between purely word-based and phrase-based representations suggests that phrasal representation alone yields no performance improvement. Moreover, experiments have shown that only small improvements in text retrieval effectiveness are obtained when using syntactical phrases to supplement the word-based representation [LeCr90].

A statistical phrase is also called an *n*-gram, referring to a sequence of *n* consecutive words in a sentence. The addition of statistical phrases to document representation demonstrates an improved classification performance [FuMR98, MlGr98]. A number of related experiments studying the effectiveness of phrases have also been conducted by others [DPHS98, CaMS01, LiAD03, BeAl04]. The results are not conclusive, and the investigation in this direction is still being actively pursued.

Term Weight Calculation

The *bag of words* approach is a simple but effective way to represent text. Formally, let the term space consist of *k* terms. A document d can be represented as a vector of term weights, $d \equiv (t_1, t_2, ..., t_k)$, where the weight t_k represents how much the kth term contributes to the semantics of document d [Seba99]. The weight can be either binary or numerical. A binary weight indicates the presence or absence of a term in a document, while a numerical weight more precisely measures the relative importance of a term in different documents. There are many ways to calculate a term's weight, and the most frequently used weighting scheme is *TF·IDF* [SaBu98]. *TF·IDF* is calculated based on *term frequency* and *document frequency*. *Term frequency* is the number of times a term *t* occurs in a document d , denoted by $TF(t,d)$. *Document frequency* measures the number of documents in which a term *t* occurs, denoted by $DF(t)$. *TF·IDF* is typically calculated as:

$$TF \cdot IDF(t,d) = TF(t,d) \cdot \log \frac{|D|}{DF(t)} ,$$

where $|D|$ is the total number of documents.

Term Space Reduction

Term space reduction is an indispensable step in text representation due to the high dimensionality of the term space. Some machine learning algorithms may not be able to handle the large number of terms, and the performance of the classification process may severely degrade. Moreover, when the number of training examples is limited, having too many terms may cause *overfitting*, the phenomenon that the classifier is tuned to learn the specific characteristics of the training data rather than the general characteristics of the categories. As a result, the classifier performs well on the training data, but much worse on unseen test data. Experiments have shown that a number of training documents roughly proportional to the number of terms is needed to avoid overfitting [Seba99]. Therefore, when the training data is insufficient, overfitting may be avoided if the number of terms is reduced. Typically the dimensionality reduction of the term space can be achieved in two ways: term selection and term extraction [Seba99].

Techniques for term selection attempt to choose a subset of terms from the original set to yield the highest effectiveness. The simplest form of term selection is to remove terms that occur less than a minimum number of times. Many sophisticated functions are also available to score terms more accurately, such as *mutual information*, *chi-square*, *information gain,* and *odds ratio*. Most of these functions take into account the relationship between individual terms and specific categories. *Chi-square* measures the dependence between a term and a category; *information gain* measures the number of bits of information obtained for category prediction with the presence or absence of a term; *odds ratio* takes into account the effect of a term in both a category and its complement. The score calculated by such functions is usually category-specific. The global score of a term can be obtained by the sum, the weighted sum, or the maximum of its local score in each individual category.

Table 2.2.1 summarizes the definitions of several common term space reduction functions. The probabilities are interpreted on an event space of documents. For example, $p(t)$ denotes the probability that a random document *d* contains term *t*. $p(c)$ denotes the probability that a random document *d* belongs to category *c*. $p(t,c)$ denotes the probability that, for a random document *d*, term *t* occurs in *d* and *d* belongs to *c*. $p(\bar{t} \mid c)$ denotes the probability that, for a document *d* that belongs to *c*, term *t* does not occur in *d*. The probabilities are estimated by counting occurrences of documents in the training set. A comprehensive study of the score functions was performed by Yang and Pederson [YaPe97]. In their report, the strengths and drawbacks of several term space reduction functions were investigated, and the experiments with K-Nearest-Neighbour

(KNN) and Linear Least Square Fit (LLSF) classifiers suggested that *information gain* and *chi-square* are the most effective score functions for term selection.

Table 2.2.1. Main functions used for term space reduction [Seba99]. t denotes a term and c denotes a category.

Function	Denoted by	Mathematical form
Information gain	$IG(t,c)$	$p(t,c)\log\frac{p(t,c)}{p(t)\cdot p(c)} + p(\bar{t},c)\log\frac{p(\bar{t},c)}{p(\bar{t})\cdot p(c)}$
Chi-square	$\chi^2(t,c)$	$\frac{\lvert D\rvert\cdot(p(t,c)\cdot p(\bar{t},\bar{c}) - p(t,\bar{c})\cdot p(\bar{t},c))^2}{p(t)\cdot p(\bar{t})\cdot p(c)\cdot p(\bar{c})}$, where $\lvert D\rvert$ is the total number of training examples
Odds ratio	$OR(t,c)$	$\frac{p(t\mid c)\cdot(1-p(t\mid\bar{c}))}{(1-p(t\mid c))\cdot p(t\mid\bar{c})}$
Mutual information	$MI(t,c)$	$\log\frac{p(t,c)}{p(t)\cdot p(c)}$
Z score	$Z(t,c)$	$\frac{p(t\mid c)-p(t\mid\bar{c})}{\sqrt{P\cdot(1-P)}\cdot(\frac{1}{\lvert c\rvert}+\frac{1}{\lvert\bar{c}\rvert})}$, where $P=\frac{\lvert c\rvert\cdot p(t\mid c)+\lvert\bar{c}\rvert\cdot p(t\mid\bar{c})}{\lvert c\rvert+\lvert\bar{c}\rvert}$, $\lvert c\rvert$ denotes the total number of examples that belong to category c

In contrast to term selection, term extraction does not necessarily use terms from the original set. Instead, it synthesizes a set of new terms based on the existing ones. Some supervised or unsupervised clustering techniques are usually applied for term extraction. Unsupervised clustered representation was first investigated by Lewis [Lewi92]. In his studies, the Reciprocal Nearest Neighbor (RNN) [Murt83] clustering analysis was used to group together semantically related words or phrases. However, his experimental results showed that both the word cluster and phrase cluster representations were inferior to their original counterparts, that is, pure word or phrase representation. Supervised clustering techniques for term extraction were later proposed by Baker and McCallum [BaMc98]. They applied distributional clustering, which took into account the associated document categories when clustering words into groups. Their experimental results showed that the term space dimensionality could be reduced by three orders of magnitude while losing only 2% accuracy.

In addition to clustering terms based on their mutual similarities, other clustering techniques, such as Latent Semantic Indexing (LSI) and Principal Component Analysis (PCA) were also applied in term extraction tasks. LSI compresses the original high dimensional term space into a lower dimensional space through matrix decomposition. It has been proved to be an effective dimensionality reduction technique [ScHP95]. Li and Jain used PCA to project data from the original term space onto a lower dimensional subspace. They showed that the performance of a Decision Tree classifier had improved with this feature extraction strategy [LiJa88].

Other preprocessing procedures, such as the removal of *stop words* and stemming can also be adopted as an effective way for dimensionality reduction. *Stop words* refer to topic-neutral words such as *articles*, *prepositions*, *conjunctions*, etc. Stemming means representing a word with its morphological root such that a group of words that share the same root can be treated as the same word. For example, different words such as *teacher*, *teach*, *teaching*, *taught* can be represented by their basic root *teach*. As a result, the number of the terms is reduced.

Other Representation Methods

In addition to the *bag of words* approach, many other representation methods have also been tried. The *Darmstadt Indexing Approach* [Fuhr85] includes statistical information about words and phrases, as well as structural information about the document, for example, the locations (e.g., in the *title*, *abstract*, or *conclusion* section) of words in the text. *WordNet* [Mill90] provides semantic information about words from a pre-defined thesaurus. Some researchers have also tried to incorporate syntactical information into text representation. For example, Ray and Craven [RaCr01] use words and phrases, as well as their syntactical categories (e.g., part-of-speech tags) to represent sentences from biomedical literature. The purpose of these representation methods is to incorporate additional information that may lead to improved classification performance but can hardly be accounted for in the standard *bag of words* approach.

2.2.2 Construction of Text Classifiers

Many statistical classification algorithms and machine learning techniques have been successfully applied to text categorization. These include methods such as Naïve Bayes [BaMc98, McNi98], Decision Tree [LeRi94, Moul96, ChHM97], Linear Discriminant Analysis (LDA) [HuPS96], Neural Networks [WiPW95, RuSr97], Logistic Regression [ScHP95], K-Nearest-Neighbour [Yang94], Linear Least Squares Fit (LLSF) [YaCh92], and Hidden Markov Models [Leek97, BiSW99, CoNT00, KuJM01, DeZa01, RaCr01, SkCR03]. Here we introduce only those methods that are directly relevant to this work, as we apply them to our classification task.

Naïve Bayes Classifier

A Bayesian classifier tries to estimate the conditional probability that an input document belongs to a category, i.e. $p(c \mid d)$, where d represents a document, and c denotes a category. We call $p(c \mid d)$ the posterior probability, which can be computed from the product of the prior probability $p(c)$ and the likelihood $p(d \mid c)$ according to Bayes theorem:

$$p(c \mid d) = \frac{p(c)\,p(d \mid c)}{p(d)}.$$

Since the probability that a document d occurs in the corpus, $p(d)$, is a fixed value for a given document *d,* we do not need to estimate it. The estimation of the posterior probability $p(c \mid d)$ is thus converted to the estimation of the prior probability $p(c)$ and the likelihood $p(d \mid c)$. If the terms of the input document are assumed to be conditionally independent given the category, the likelihood $p(d \mid c)$ can be simply calculated by multiplying the likelihood of category *c* with respect to each term:

$$p(d \mid c) = \prod_{k=1}^{|T|} p(t_k \mid c),$$

where t_k is the weight of the k$^{\text{th}}$ term in document d, and $|T|$ is the total number of terms. The probability distributions $p(c)$ and $p(t_k \mid c)$ are usually assumed to have known parametric forms, and the learning task is essentially the estimation of the parameters (See [DuHa73] for more information).

Classifiers that are based on the above independence assumption are called Naïve Bayes classifiers, and they are often applied to text categorization tasks [LeRi94, ScHP95, BaMc98, McNi98]. However, the independence assumption is often violated in practice.

For example, the probability that the word *learning* occurs will be largely increased if it is preceded by the word *machine*. To capture the probabilistic dependencies between terms, the application of Bayesian Networks was studied [Saha96, ChHM97]. The major limitation of Bayesian Networks is that learning the model quickly becomes intractable as the number of terms grows [Saha96]. Considering the high dimension of the term space and the relatively limited amount of training data for most text classification tasks, it is very hard to learn a full, unrestricted Bayesian Network model without incorporating much prior knowledge.

Support Vector Machines

Support Vector Machines (SVMs) were introduced into the area of text categorization by Joachims [Joac98], and has become a state-of-the-art method in this field [DPHS98, Joac98, Joac99, YaLi99]. SVM algorithms map the training examples into a high dimensional feature space based on a kernel function, and try to find a hyperplane to separate the mapped points, such that the margin between positive and negative examples is maximized and the number of misclassifications is minimized. The examples closest to the hyperplane are called support vectors. These examples essentially determine the position of the hyperplane.

As an example, we consider a binary linear SVM. Let x be a data point, w be a weighting vector, and b be a constant. The hyperplane for a linearly separable space can be defined by a linear function:

$$f(x) = wx + b,$$

where *wx* + *b* > 0 for positive data and *wx* + *b* < 0 for negative data. We use y_i to denote the label of data point x_i, with the value 1 for positive data and –1 for negative data. In order to minimize the number of misclassifications, $f(x)$ must satisfy the condition:

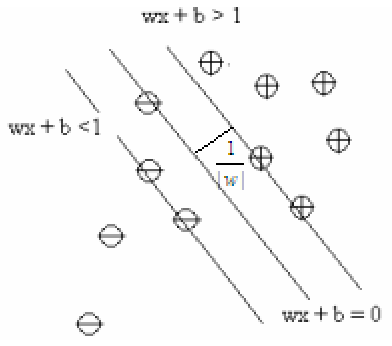
$$y_i f(x_i) \geq 0 \text{ for } i = 1,...,n.$$

This general principle is called *empirical risk minimization*. However, SVM emphasizes the confidence in the classification more than the number of misclassifications, that is, the mapped data in the two half spaces should be far away from the hyperplane. This can be done by further restricting the function $f(x)$ to satisfy the condition:

$$y_i f(x_i) \geq 1 \text{ for } i = 1,...,n,$$

namely, *wx* + *b* > 1 for positive data and *wx* + *b* < 1 for negative data. In such a case, the margin between the two half spaces is of width $\frac{2}{|w|}$. To achieve large confidence, the margin should be as wide as possible. Figure 2.2.1 illustrates the hyperplane found by the linear SVM algorithm to separate positive and negative data examples.

Figure 2.2.1. SVM uses a hyperplane to separate positive and negative examples. The hyperplane h is defined by the linear function f(x) = wx+b = 0. The examples close to h (distance from h equal or less than $\frac{1}{|w|}$ *) are called support vectors.*



So far, the solution to SVM is converted into a constrained optimization problem, that is, maximizing the margin $\frac{2}{|w|}$ under the constraints $y_i f(x_i) \geq 1$ for $i = 1,...,n$. This constrained optimization problem can be solved by introducing Lagrange multipliers. We refer readers to a review by Vert *et al.* [VeTS04] for the detailed procedures of finding the optimal parameters *w* and *b*.

According to Joachims [Joac98], SVMs outperform other classifiers, such as Naïve Bayes, K-Nearest-Neighbour, and Decision Tree, in text categorization. In addition, they have some important advantages over other text classifiers. As mentioned in Section 2.2.1, text categorization is characterized by the high dimensional feature space. Joachims [Joac98] has pointed out that in many text categorization tasks, only a few features are irrelevant among the large number of features. In such cases, feature selection may hurt the classification performance. Hence, a good classifier should be able to combine a large number of features. Moreover, the high dimensional feature space may lead to a very sparse document representation, i.e. only a few entries in the feature vector are non-zero. Sparse data representations typically degrade classification performance, since little information is provided for the classifier to learn useful statistics for most of the features. Kivinen *et al.* [KiWA97] have shown, from both theoretical and empirical perspectives, that algorithms that have similar inductive bias[1] to SVMs are well suited for the classification tasks where the dimensionality of the feature space is high and the data representations are sparse. SVMs were initially designed to handle binary classification

[1] The inductive bias of a machine learning algorithm refers to the hypotheses or assumptions, which are generated from the training examples by the learner and will be used to predict the output values for future unknown examples.

problems, but the algorithms have since been extended to deal with multi-class [CrSi01], hierarchical [CaHo04], and ranked [Joac02] classifications.

Maximum Entropy

The Maximum Entropy method was introduced for text classification by Nigam *et al.* [NiLM99]. The basic principle of Maximum Entropy is that without prior knowledge, the least informative distribution, i.e. the distribution with the maximum entropy, is preferred. In text classification tasks, like Bayes classifiers, Maximum Entropy classifiers estimate the conditional probability of the class label given the document, that is, $p(c \mid d)$, where d represents an input document, and c denotes a category.

In Maximum Entropy classifiers, the training data is used to set constraints on the conditional distribution $p(c \mid d)$. According to Nigam *et al.* [NiLM99], if we define any real-valued function of the document and the class to be a feature, and restrict the distribution $p(c \mid d)$ to have the expected value for this feature as derived from the training data, then a unique distribution $p^{*}(c \mid d)$ with maximum entropy will always exist and conform to the exponential form.

More formally, if |*D*| denotes the number of training examples, *d* denotes a document, *c* denotes a category, $c(d)$ denotes the true category of document *d*, and f_i denotes a feature function, we restrict the conditional distribution to satisfy the constraint:

$$\frac{1}{|D|}\sum_{d \in D} f_i(d, c(d)) = \sum_{d \in D} P(d) \sum_{c \in C} P(c \mid d) f_i(d, c) .$$

The left part of the above equation represents the expected value of the feature f_i derived from the training examples, and the right part represents the expected value of the feature

f_i calculated based on the distribution $p(c \mid d)$. In such a case, among all the distributions $p(c \mid d)$ that satisfy the given constraints, the optimal distribution $p^*(c \mid d)$ with the maximum entropy exists and conforms to the exponential form:

$$p^*(c \mid d) = \frac{1}{Z(d)} \exp\{\sum_i \lambda_i f_i(d,c)\},$$

where f_i is a feature, λ_i is the corresponding parameter to be estimated, and Z is the normalizing factor, defined as $Z(d) = \sum_{c \in C} \exp\{\sum_i \lambda_i f_i(d,c)\}$.

Many studies have been conducted on applying Maximum Entropy to text classification [BSAG98, NiLM99, ZJXG05], as well as to other natural language processing problems [RaRR94, Ratn96, McFP00]. The experimental result by Nigam shows that the Maximum Entropy classifier outperforms the Naïve Bayes text classifier [NiLM99]. In contrast to Naïve Bayes classifiers, Maximum Entropy classifiers do not make any independence assumptions about features. In addition, the computational complexity of the parameter estimation for Maximum Entropy classifiers is lower, since it requires merely differential calculus techniques or gradient search procedures to determine the best parameters [Berg97], while Bayesian learning usually needs complex multidimensional integration [DuHa73]. We will discuss Maximum Entropy classifiers in detail in Chapter 5.

2.2.3 Performance Evaluation

The performance of a text classifier is typically measured by its effectiveness, that is, its ability to make the right classification decision. In this section, we discuss the performance measures for flat text categorization and hierarchical text categorization.

Performance Measures for Flat Categorization

The most commonly used measures for text categorization are *Precision* and *Recall* [Seba99]. *Precision* of a classifier, for a category *c*, is the ratio of documents correctly classified under *c* with respect to all the documents assigned to *c* by the classifier. *Recall* measures the ratio of documents correctly classified to *c* with respect to all the documents that should be classified to *c*. We define TP_c, *True Positive*, as the number of documents correctly classified into category *c*; FP_c, *False Positive*, as the number of documents incorrectly classified into *c*; TN_c, *True Negative*, as the number correctly rejected from *c*; and FN_c, *False Negative*, as the number incorrectly rejected from *c*. The *Precision* and the *Recall* of category *c* are then defined as:

$$Precision_c = \frac{TP_c}{TP_c + FP_c}, \quad Recall_c = \frac{TP_c}{TP_c + FN_c}.$$

Based on the local *Precision* and *Recall* with respect to each category, the global *Precision* and *Recall* over the whole category space can be calculated in two ways: *microaverage* and *macroaverage* [Seba99]. *Microaverage* gives each document equal weight, while *macroaverage* gives each category equal weight:

$$\text{Microaverage:}\quad Precision_{mi} = \frac{\sum_{c \in C} TP_c}{\sum_{c \in C} (TP_c + FP_c)}, \quad Recall_{mi} = \frac{\sum_{c \in C} TP_c}{\sum_{c \in C} (TP_c + FN_c)};$$

$$\text{Macroaverage:}\quad Precision_{ma} = \frac{\sum_{c \in C} \frac{TP_c}{TP_c + FP_c}}{|C|}, \quad Recall_{ma} = \frac{\sum_{c \in C} \frac{TP_c}{TP_c + FN_c}}{|C|}.$$

However, neither *Precision* nor *Recall* can be an accurate indicator of effectiveness in isolation from each other. Usually high *Precision* can be obtained at the price of low *Recall* and vice versa [Seba99]. Hence, a classifier should be evaluated by a measure which combines both. The most popular combination is called *F-function* or *F-measure* [VanR79], which balances the relative degree of importance of *Precision* and *Recall*, and is defined as:

$$F = \frac{(\beta^2 + 1) * Precision * Recall}{\beta^2 * Precision + Recall} \quad , \quad 0 \le \beta \le +\infty \, .$$

The parameter β adjusts the weight assigned to *Precision* and *Recall*, and the most common values of β are 0.5, 1.0, and 2.0. When β is 0, the value of *F-measure* is the same as *Precision*; when β tends to ∞, the value of *F-measure* tends towards *Recall*.

Performance Measures for Hierarchical Text Categorization

Performance measures for flat text categorization do not take into account inter-category relationships. If the set of categories is hierarchical by nature, flat measures do not reward *partial success* when a document is correctly assigned to parent categories but not to child categories, nor do they differentiate misclassifications between close categories and totally unrelated categories in the category hierarchy. To address this issue, some frameworks specifically designed for hierarchical text categorization have been proposed. For example, Sun *et al.* [SuLN03] proposed two measures: C*ategory Similarity measure* and *Distance-based measure*. Here we give the brief concepts, and refer readers to the original paper for the detailed definitions of the two measures.

Category Similarity measure gives more weight to misclassifications into similar categories than into totally unrelated categories. The similarity between categories, which can be defined manually or calculated based on the features of the categories, measures their semantic closeness. The contribution of the misclassification is calculated based on how similar the correct and assigned categories are in comparison to the average category similarity.

Distance-based measure evaluates the classification performance based on *distance*, formally the shortest path (measured by the number of edges), between two categories in a category tree. To calculate the contribution of the misclassified documents, a threshold distance, Dis_t, is first defined by the user. If the distance between two categories is smaller than Dis_t, the contribution of the misclassification between them would be positive; if the distance is equal to Dis_t, the contribution would be zero; if the distance is greater than Dis_t, the contribution would be negative.

Other performance measures have also been proposed. For example, a method that calculates *Precision* and *Recall* from the intersection of the predicted category set and the true category set is proposed by Kiritchenko *et al.* [KiMF05]. For any document belonging to a set of categories CS_i but classified into a set of categories CS_k, the original category sets are extended with the corresponding ancestor categories:

$$\hat{C}S_i = \{ \bigcup_{c \in CS_i} Ancestors(c) \} \ , \ \hat{C}S_k = \{ \bigcup_{c \in CS_k} Ancestors(c) \} \, .$$

The hierarchical *Precision* and *Recall* can then be calculated as:

$$Precision = \frac{\sum_i | \hat{C}S_i \cap \hat{C}S_k |}{\sum_i | \hat{C}S_k |} \ , \quad Recall = \frac{\sum_i | \hat{C}S_i \cap \hat{C}S_k |}{\sum_i | \hat{C}S_i |} \, .$$

The performance measure proposed by Kiritchenko *et al.* can also be applied to flat multi-label classification tasks to reward the partially correct assignment of the label set. A comprehensive study in the performance evaluation of hierarchical text categorization can be found in the doctoral dissertation of Kiritchenko [Kiri05].

2.3 Previous Research on Sentence Categorization

As mentioned in Section 2.1.4, sentence level text categorization has been applied to many tasks, such as entity and relationship extraction, automatic text summarization, rhetorical analysis, and others. We next discuss in detail the applications that are related to our work.

2.3.1 Named Entity and Relation Extraction

Named entity and relation extraction has been a rapidly growing area in recent years. Bikel *et al.* [BiSW99] developed a statistical approach to find names and other entities in sentences using a variant of the standard Hidden Markov Model (HMM). A named entity recognition system built on the Maximum Entropy framework was presented by Borthwick *et al.* [BSAG98] to address similar recognition problems. In the biomedical domain, recognizing named entities such as genes, proteins, cells, as well as their interactions, is generating increasing interest. Work on the identification of biomedical named entities has been done by many authors [CoNT00, YKKM03, LeHR03].

In the field of entity relation discoveries, hidden Markov models (HMMs) are often used [Leek97, RaCr01, SkCR03]. HMMs process the input text as a sequence of words rather than a *bag of words*. Hence, they can handle structural constraints as well as the statistics of words. Methods based on the statistical analysis of co-occurring terms in phrases, sentences, or abstracts [CrKu99, DBNW02], and on pattern or template matching using a set of rules [OHTT01, KaMO04] are also explored to search for interactions among genes or proteins.

However, the successful recognition of an entity name or the exact matching of a relation pattern does not assure the reliability of the information. Additional knowledge about the level of belief in the extracted information can be helpful to improve the performance of certain information extraction tasks.

2.3.2 Automatic Abstract Generation

Automatic abstract generation provides a quick mechanism for users to obtain the main points of an article without having to read the full text. Extensive experiments of applying text categorization to automatic text summarization have been performed. For example, Kupiec *et al.* [KuPC95] used a Naïve Bayes classifier to rank sentences by the order of summarization quality. The features they used were first proposed by Luhn [Luhn58] and Edmundson [Edmu68], and include thematic word frequencies, sentence length and location, the presence or absence of cue phrases (e.g. *in conclusion, this article*) or title/header words. Similar approaches were adopted by others [GKMC99, NSMU01, NSUI02], among which, Goldstein *et al.* [GKMC99] incorporated query-relevant information to generate query-specific rather than generic summaries.

Other automatic text summary approaches such as MEAD [RJBT00], and MMR-MD [Gana02], initially cluster passages or documents by topics, and then select sentences based on their similarity to the centroid of the cluster and their dissimilarity to the already selected sentences. The principal merits of such cluster-based summarization lie in its capability of generating summaries for multiple documents, as well as to reduce the redundancy between selected sentences.

2.3.3 Rhetorical Relation Analysis

Research on categorizing rhetorical relations among text units has become fairly common lately [KnSa98, ChYa00, MaEc02, LiBr04]. Rhetorical relations (also called discourse relations, or coherence relations) can be described as a set of relations that link one text unit to another, making the text coherent. Some typical relations are *evidence*, *contrast*, *concession, explanation*, etc. That is, one unit of text can provide evidence for the unit preceding it, raise a contrast to it, or further explain it. Studies show that the recognition of rhetorical relations can facilitate the processing of text and speed reading time.

Analysis of the discourse relations in a scientific article has been performed by a number of researchers. Mann and Thompson [MaTh88] proposed *Rhetorical Structure Theory* (RST) with a catalogue of twenty-three relations (e.g. *purpose*, *cause*, *elaboration*), which had a particular influence on most of the later work. Sanders *et al.* [SaSN92] presented a method to classify the relations among sentences in terms of cognitive relevance (e.g. *causal* or *additive*, *semantic* or *pragmatic*, *positive* or *negative*). Knott *et al.* tried to classify relations on the basis of cue phrases (e.g. *because*, *but*, *and*) and built a taxonomy of relational phrases [KnDa94, KnSa98]. However, both methods relied on the analysis of human experts from a linguistic perspective. With a much coarser level of granularity: *contrast*, c*ause-explanation-evidence*, *condition*, and *elaboration*, Marcu and Echihabi [MaEc02] successfully applied a machine learning method to the classification of rhetorical relations between sentences. Their experiments were conducted on two corpora, one consisting of 41,147,805 sentences and another consisting of 1,796,386 sentences. A Naïve Bayes classifier was built for each rhetorical relation pair, and the recognition accuracy of some relations was as high as 93%.

Rhetorical relations were used in automatic abstract generation by Chuang and Yang [ChYa00]. Sentences were first broken into segments by special cue markers [Marc98] (e.g. *because*, *but*, *if*, etc.). Next, each segment was represented by the set of features introduced by Kupiec *et al.* [KuPC95], as well as by rhetorical relations in the *Rhetorical Structure Theory* [MaTh88]. Then machine learning algorithms, including Decision Trees, Naïve Bayes, and Neural Networks [YaPH99], were applied to classify segments based on whether they should (or should not) be included in the abstract. On a dataset of nine U.S. patents, where the number of segments ranged from 19 to 139, the highest accuracy obtained was 78%, better than the commercial software Microsoft Word Summarizer, which had an accuracy of 60.8%.

Rhetorical relation analysis can also be adopted to assist in information extraction tasks in order to extract the desired information more effectively and accurately. An attempt to further classify named entity relations according to a set of rhetorical relations has been reported by Light and Bradshaw [LiBr04].

2.3.4 Rhetorical Zone Analysis

The study of categorizing sentences according to different rhetorical zones, such as *Background*, *Problem*, *Method*, and *Result*, was first motivated by the attempt to generate user-customizable abstracts. For example, abstracts aimed at novice readers typically provide an overview of the field and a basic introduction of the author’s work, while abstracts oriented towards experienced readers may focus on the detailed problem and methodology. Teufel and Moens extended the work of Kupiec *et al.* [KuPC95] by further classifying the extracted sentences into seven categories according to their rhetorical zones: *Background*, *Topic*, *Related Work*, *Purpose and Problem*, *Solution and Method*, *Result*, and *Conclusion* [TeMo97, TeMo99, TeCM99]. In a corpus of 201 articles, they successfully extracted 64% of the abstract-worthy sentences, and subsequently assigned the right rhetorical zones to 64% of the correctly extracted sentences using a Naïve Bayes classifier.

Similar work was also done by others. A study of categorizing sentence types from Medline abstracts was conducted by McKnight and Srinivasan [McSr03]. In their experiment, 7,253 abstracts of *Randomized Controlled Trials* from Medline were broken into sentences and each sentence was labeled as one of the four types: *Introduction*, *Method*, *Result*, and *Conclusion*. A Support Vector Machine (SVM) model was trained and evaluated on cross-validation data, and high classification accuracy (average *F-measure* of 85%) was obtained.

Based on Teufel and Moens's flat structure of rhetorical zones, Mizuta and Collier proposed a zone analysis scheme with shallow nesting [MiCo04a, MiCo04b, MuMC05]. They treat sentence segments in biology articles as the basic classification units. The set of rhetorical zones is divided into three groups: (1) *background* information, *problems* to be solved, and the author's own work including *method*, *result*, *insight*, *implication*, and e*lse*; (2) *connection* or *difference* between findings; (3) *outline* of the paper. Zones can be nested, that is, a sentence segment may simultaneously fit into multiple zones. For example, *connection* or *difference* zone typically overlap with another zone such as *insight* and *implication*. Two classification methods, Naïve Bayes and Support Vector

Machine, were used to build a set of binary classifiers for each rhetorical zone. On a dataset consisting of 3,637 sentences from 20 journal articles in molecular biology, an overall *F-measure* of 70% was obtained over all zones. The recognition of some specific rhetorical zones, for instance, *Method*, yielded an *F-measure* as high as 87%. Nevertheless, the sophisticated scheme made the differentiation among certain zones difficult. Consequently, the predictive accuracy for zones such as *connection*, *difference*, *insight*, and *implication* were relatively low, with an *F-measure* lower than 50%.

2.3.5 Other Applications

Many other interesting applications for sentence level classification have been investigated. Mercer and DiMarco studied the automatic classification of relationships between the citing and cited paper, such as *contrast*, *supportive*, and *corrective*. Their work showed that rhetorical relations, and discourse cues such as *not*, *previously*, *although*, *in order to*, as well as hedging cues, that is, linguistic expressions that qualify the confidence in a proposition such as *perhaps*, *might*, *demonstrate*, play an important role in automatic citation analysis [MeDi03, MeDK04].

Light *et al.* [LiQS04] tried to explore the use of speculative language in Medline abstracts. Results from manual annotation experiments suggested that speculative sentences in the context of bioscience literature could be reliably annotated by human experts. In addition, they built an SVM text classifier to distinguish speculative sentences from definite ones. On a data set of 1,629 sentences, the *Precision* and the *Recall* were 84% and 39% respectively. Their preliminary results showed that reliable automatic methods might also be developed.

A more comprehensive study of the semantic patterns of a sentence was conducted by Friedman *et al.* [FAAC94]. They developed a medical text processor to identify clinical information in radiology reports and mapped the information into a structured representation containing controlled vocabulary terms. In their system, the original sentence from radiology reports was translated into a structured form according to certain predefined semantic grammar rules. Then single words or multiword phrases were mapped into some semantic categories such as *Negation* (e.g. *no evidence*), *Certainty* (terms affecting the certainty of a finding, e.g. *possible*, *appears*), *Change* (terms defining a change in findings where a change is an improvement or worsening of a finding, e.g. *decrease*, *improving*), *Degree* (terms denoting the severity of a finding, e.g. *mild*, *moderate*, *severe*), *Cfinding* (terms denoting a complete radiology finding, e.g. *cardiomegaly*, *pleural effusion*) etc. Evaluation on a dataset of 230 radiology reports showed that the system achieved promising performance in terms of *Precision* and *Recall* (both 85%). However, the method was limited to several clinical domains. It is hard to extend it to more general fields where the language structure is more complex and the controlled vocabulary may not be well defined.

Chapter 3

Overview of Multi-dimensional Fragment Classification

In this chapter, we give an overview of multi-dimensional fragment classification. We first introduce the annotation scheme proposed by Shatkay *et al.* [ShWR06, WiRS06], under which the multi-dimensional fragment classification is performed. We next briefly introduce the subtasks of automatic fragment annotation and the issues this work addresses. To illustrate the use for this type of fragment classification we conclude the chapter with a discussion of several potential applications for our work.

3.1 Fragment Annotation: An Introduction

To identify information-bearing text units within scientific literature, a set of categories are defined to characterize the text that satisfies various types of information needs. The categories are grouped into five dimensions, defined as follows [ShWR06]:

- *Focus*: Distinguishes whether the text unit describes *Scientific* discoveries or findings, *Methodology* for some experiments, or *Generic* information such as general state of knowledge or the organization of the paper.
- *Polarity*: Indicates whether an assertion is stated *Positively* or *Negatively*.
- *Certainty*: Measures the degree of confidence regarding the validity of an assertion. A scale from 0 to 3 is used to measure *Certainty*. The lowest degree (0) represents *Complete uncertainty*, that is, the statement explicitly expresses there is an uncertainty or lack of knowledge. The highest degree *Complete*

certainty (3) represents a known or proven fact. The intermediate degrees (1 and 2) represent *Low certainty* and *High likelihood* respectively.

- *Evidence*: Indicates whether an assertion is supported by evidence. The types of evidence are defined as follows:
  - *No evidence*: There is no indication of evidence, denoted as *E0*.
  - *Claim of evidence without verifying information*: There is a claim of evidence, but no explicit verifying information is provided, denoted as *E1*.
  - *Explicit citation*: Explicit citations are made to support the assertion, denoted as *E2*.
  - *Explicit evidence*: Evidence is provided in the form of reference to experiments reported within the body of the paper, denoted as *E3*.
- *Direction/Trend*: Indicates whether an assertion reports an *Increase* or a *Decrease* in a specific phenomenon or activity. It captures the semantic meaning of the observed phenomenon itself, in contrast to the *Polarity* dimension, which defines the direction of an assertion from the syntactical perspective.

The process of categorizing a text unit along the five dimensions is called *annotation*. We can use a single tag to represent the category labels of a text unit along the five dimensions. The basic annotation unit is defined as a fragment within a sentence, because a paragraph or a sentence is typically too heterogeneous in contents to be characterized by one single tag. The fragmentation of a sentence occurs at the point where there is a change in any of the five dimensions defined above. Here we give two examples of fragmentation, with each fragment annotated by a tag. A tag consists of a sequence of numbers and letters, denoting *Fragment Number*, *Focus* (*S*, *G*, *M*, or a combination of them), *Polarity* (*P* or *N*), *Certainty* (*0 – 3*), *Evidence* (*E0 – E3*), and *Trend* (+: *Increase* or -: *Decrease*) [ShWR06].

> Furthermore, we show that the increased somal [Ca2+]i ***1SP3E3+*
> and decreased cell survival following proximal transactions ***2SP3E0-*
> are not due to less frequent or slower plasmalemmal sealing or Ca2+ entry
> through plasmalemmal Na+ and Ca2+ channels. ***3SN3E0-*

The sentence is fragmented into three parts to reflect the changes in *Trend*, i.e. from an *Increase* fragment (fragment 1 as specified by "*increased somal*") to a *Decrease* fragment (fragment 2 as specified by "*decreased cell survival*"), as well as the changes in *Polarity*, i.e. from a *Positive* fragment (fragment 2) to a *Negative* fragment (fragment 3).

> Another potential factor in promotion of apoptosis, inducible NO synthase (37), ***1SP2E2+*
> is limited in distribution to perivascular infiltrates at the peak of inflammation ***2SP3E0*
> and is unlikely to contribute to widespread neuronal loss. ***3SP1E0-*

In the above sentence, fragmentation is motivated by the changes in *Certainty*, i.e. from *High likelihood* ("*potential factor*" in fragment 1), to *Complete certainty* ("*is limited in*" in fragment 2), to *Low certainty* ("*unlikely*" in fragment 3), as well as the changes in *Evidence* and *Trend*.

To better demonstrate the annotation criteria, Table 3.1.1 provides an example for each category along each dimension. All examples are cited from the Annotation Guidelines [ShWR06], to which we refer users for more details.

Table 3.1.1. Annotation examples.

Annotation Type: Focus	Category
We demonstrate that ICG-001 binds specifically to CBP.	Scientific
DNA sequence was collected and analyzed on an ABI Prism 377 automated DNA sequencer.	Methodology
To deal with them, the world needs to reformulate the biomolecular paradigm that has been exploited in the last two centuries.	Generic
Annotation Type: Polarity	Category
She2p forms a stable dimer in solution.	Positive
None of the NBD rats had classic Borna disease or meningoencephalitis.	Negative
Annotation Type: Certainty	Category
We sought to establish whether or not She2p dimerization is required for RNA binding.	0: Complete uncertainty
Partial inhibition of this attachment indicated that other pathways might also exist.	1: Low certainty
Reports of Purkinje and granule cell loss in Cblm (16) suggest overlap with this neonatal infection paradigm.	2: High likelihood
We determined TP53 gene mutation in two cases and the genome-wide allelotype, AXIN1, and CTNNB1/beta-catenin gene mutation in one case.	3: Complete certainty
Annotation Type: Evidence	Category
ICG-001 has no effect on AP1 and CRE reporter constructs.	0: No evidence
At the present time, then, the available data would support the notion that b-*catenin* mutations are only rarely seen in sporadic colon cancer.	1: Claim of evidence without verifying information
Neonatally infected rats are reported not to have inflammation (6–10).	2: Explicit citation
Astrocytosis and microgliosis were evident in all brain regions by 3 wk p.i. (Fig. 7).	3: Explicit evidence
Annotation Type: Trend	Category
We show that treatment with ICG-001 induces apoptosis in colon carcinoma cells.	+: Increase
ICG-001 selectively blocked the beta-catenin-CBP interaction.	-: Decrease
Several lines of evidence demonstrate that *I. scapularis* TROSPA is a specific ligand for *B. burgdorferi* OspA.	NA

The feasibility and reliability of the fragment annotation were verified by a formal preliminary test [WiRS06], in which 70-80% inter-annotator agreement was obtained among twelve independent annotators on a set of 101 sentences, suggesting the annotation criteria are well defined and can be followed consistently by human annotators.

3.2 Multi-dimensional Fragment Classification: Toward Automatic Annotation of Text Fragments

Our general goal is to automatically annotate sentence fragments. The task of automated fragment annotation can be divided into three subtasks. First, a sizable training corpus is manually annotated under the Annotation Guidelines [ShWR06]. This task is currently in its final stages[1]. In this thesis we use about 2,000 annotated sentences, and the final corpus will consist of 10,000 annotated sentences from the biomedical literature. Second, we build text classifiers on the current manually annotated corpus and automatically classify each fragment along the five dimensions defined above. We evaluate classification performance on the available part of the corpus. Finally, we plan to extend the work to automatically process raw documents, that is, automatically breaking sentences into fragments, and using our classifier to annotate each fragment according to the predefined criteria. This thesis focuses only on the second step, that is, training text classifiers on the manually annotated data and evaluating the performance of automatic fragment classification along the five dimensions, without considering the issue of

[1] The data used in the experiment part of this thesis consists of parts of this annotated corpus.

breaking documents and sentences. We refer to this work as the multi-dimensional fragment classification.

Since we are trying to classify fragments along five dimensions where the classification definitions vary greatly, we choose different data representations, classification algorithms, and evaluation methods for different dimensions. We follow the well-defined procedures of text categorization: text preprocessing, data representation, classifier construction, and performance evaluation [Seba99], described in Section 2.2. We discuss these procedures in detail in chapters 4-6 respectively.

3.3 Applications of Multi-dimensional Fragment Classification

The annotation based on the multi-dimensional fragment classification allows for a user-customizable scoring scheme to calculate the utility of a fragment, and consequently enables the identification of high-utility (information-bearing) fragments from a document, as well as further differentiation among various types of important fragments.

The fine definition of categories allows a comprehensive scoring scheme to measure the utility of a fragment from a variety of perspectives. A typical view of high-utility text is a statement discussing scientific discoveries or methods with high level of confidence and evidence. Therefore, typically a higher score may be assigned to the *Scientific* and *Methodology* categories than to the *Generic* category, and the scores for the categories along the *Certainty* and *Evidence* dimensions will increase with the level of confidence or evidence. The scoring scheme can also be customized by the user to highlight certain categories of special interest. For instance, a user may want to investigate assumptions and conjectures in a document. In this case, the certainty levels of

Low certainty and *High likelihood* should be assigned high scores. If a user wants to focus on the authors' own experimental procedures or results, the *Evidence* degree *Explicit evidence* will receive a high score. Alternatively, if one is interested in certain biological phenomena, such as biological interactions, a *Decrease* or an *Increase Trend* will be assigned a high score.

After assigning different scores to different categories, a fragment can be scored based on its annotation. The fragments with top ranking scores can be selected as bearing high utility for further processing. To better pinpoint the information that meets a user's requirements, he/she can further differentiate among various types of information-bearing text units. For example, a user may want to separate experimental methods from scientific facts, or distinguish the author's own discovery from existing work.

There are several potential applications of the fragment annotation and the identification of high-utility fragments.

First, the classification and the identification of high-utility fragments can be applied to automatic summarization. We can simply generate an abstract (summary) by selecting the fragments with top ranking utility scores within a document. We can also tailor the abstracts to best satisfy the requirements of the user. For example, the abstract can focus on a special category of fragments, such as experimental methods, experimental results or related work.

Second, the identification of high-utility fragments can improve the performance of document categorization. We can classify documents based on the importance of sentences. To highlight the essential parts of a document, we can assign high weight to words or phrases occurring in fragments with high scores or in fragments of certain category. For example, we can attribute more weight to words occurring in fragments whose *Evidence* level is *Explicit evidence*, namely, the author's own experimental results, than to words occurring in fragments whose *Evidence* level is *Explicit citation*, that is, others' work. We can also classify documents based only on the high-utility sentences, and ignore the sentences that are less informative. Previous experiments have shown that assigning different weights to individual sections [HaRL05], or basing classification on selected passages [BrSC05] can improve predictive accuracy. We expect that assigning less weight to – or completely excluding – insignificant sentences can help to accurately identify the main points of a document and consequently improve the performance of document categorization.

Third, the identification of information-bearing fragments can improve the quality of certain information retrieval tasks, where documents from a corpus are ranked to retrieve the most relevant documents. Common experience shows that, when performing keyword search using a search engine, such as Google or PubMed, a large number of documents may be returned, and typically only the top ranking ones will be chosen for further analysis. Therefore, it is important to first return the most relevant documents to the user. To address this issue, we can use the fragment scores calculated based on the annotation to post-process the retrieved document. Documents can then be ranked based on the overall score of the fragments within them, and those documents with high scores will be returned first.

We may also integrate the process of information retrieval with the identification of important fragments. For example, documents with keywords occurring frequently in high-utility sentences can be considered as more relevant than those with keywords uniformly distributed over the whole documents.

Last, the annotation of fragments can increase the reliability of certain information extraction tasks, such as relationship extraction [WiRS06]. Most existing work on relationship extraction is based on the statistical analysis of co-occurring words or phrases, or on pattern or template matching using a set of rules [BAOV99, CrKu99, HuDG00, OHTT01, DBNW02, KaMO04]. However, the presence of a gene name, the co-occurrence of a protein name and a subcellular location, or even the exact matching of a pattern such as *A interacts with B* in the text, does not necessarily assure the reliability and value of the information. With the annotation indicating the statement is in the affirmative, as well as its high *Certainty* and *Evidence* level, we can make sure that the extracted fact is more accurate and reliable.

In summary, multi-dimensional fragment classification provides a comprehensive description of each individual fragment and enables the substantiation of knowledge at the sentential level, which is likely to serve a variety of applications in the biomedical research community.

Chapter 4

Text Preprocessing and Representation

As surveyed in Section 2.2.1, in order to map the free-form text into a format that is interpretable by common machine learning algorithms, several preprocessing procedures need to be applied. We divide the preprocessing procedures into two steps: term formation and term space reduction. Term formation refers to mapping free-form text into a vector of terms, while term space reduction refers to selecting or extracting a smaller set of terms from the original term set. We next examine these two stages closely.

4.1 Term Formation

In Section 2.2.1, we have mentioned that typically terms are formed by single words or phrases, including statistical phrases and syntactical phrases. To map raw text to a fixed-length vector of terms, we consider three types of preprocessing: tokenization and part-of-speech (POS) tagging, statistical phrase generation, such as *n*-gram generation, and syntactical phrase generation, namely, text chunking.

4.1.1 Tokenization and Part-of-Speech Tagging

Tokenization means breaking a string of characters into a sequence of words, delimiters, and whitespace characters (spaces, tabs, and line breaks). Words and delimiters are called tokens, and whitespace characters are treated as boundaries. A word can consist of a series of letters, digits, or special characters (e.g. -, _). Part-of-speech (POS) tagging attempts to assign a syntactical category such as *noun*, *adjective*, or *verb*, to each token.

We use *Medpost* [SmRW04], a POS tagger based on a hidden Markov model (HMM), to perform tokenization and POS tagging. Medpost first maps each sentence to a sequence of tokens based on a set of Perl regular expressions, then passes the tokens into the HMM model which outputs the most likely tag sequence. Table 4.1.1 is the output of the tokenization and POS tagging for the sentence "*ICG-001 selectively blocked the beta-catenin-CBP interaction without interfering with the beta-catenin-p300 interaction*".

Table 4.1.1. An example of tokenization and part-of-speech (POS) tagging by Medpost. The syntactical categories of the POS tags can be found in Table 4.2.1.

ICG-001	NN
selectively	RR
blocked	VVD
the	DD
beta-catenin-CBP	NN
interaction	NN
without	II
interfering	VVG
with	II
the	DD
beta-catenin-p300	NN
interaction	NN
.	.

The MedPost tag set consists of 60 POS tags listed in Table 4.2.1. We choose Medpost as the tokenizer since it is especially designed for biomedical literature. Medpost was trained on a corpus of 5,700 manually tagged sentences from Medline, and achieves over 97% accuracy [SmRW04].

4.1.2 *n*-gram Generation

An *n*-gram is defined as a sequence of *n* consecutive words in a sentence fragment. *n*-grams aim to capture the characteristic co-occurrences of words. The most straightforward way to generate word sequences of length up to *n* (including single words, bigrams, … , (*n-1*)-grams, and *n*-grams) is described as follows. The tokens of a fragment are sequentially read and pushed into a queue. When *stop words*, delimiters, or fragment boundaries are encountered, word sequences of size *1* to *n* are generated from the front of the queue; then the front is popped, word sequences of size *1* to *n-1* are generated; and the process repeats until the queue is empty. After all the word sequences are generated, based on the training corpus, their frequencies are calculated and only those occurring more than a minimum number of times are retained. We refer to this process as the *basic* approach.

However, the arbitrary combination of words in the above approach leads to a tremendous number of terms. To avoid the dramatic growth of the term space, several algorithms have been introduced. Some interleave *n*-gram generation with the removal of terms occurring less than a minimum number of times [MlGr98, Furn98]; others integrate *n*-gram construction with term selection using term space reduction functions [BeAl04]. The underlying idea of these optimization algorithms is to form *n*-grams from the (*n-1*)-grams that have already been selected as candidate terms according to certain predefined criteria. We implement both the *basic* algorithm, which generates all *n*-grams first then removes less frequent ones, and the method that discards rare (*n-1*)-grams first and generates *n*-grams based on the reduced set of (*n-1*)-grams [MlGr98].

To capture a longer word sequence, *stop words* in the word sequence can be ignored. For instance, the fragment, "*the response to stress will be measured through behavioral observation*", can be represented as a *5*-gram, "*response stress measured behavioral observation*", when *stop words* are ignored. Otherwise, if we break the word sequence at every *stop word*, the longest sequence is the bigram "*behavioral observation*". In our work, we investigate the classification performance based on both representations.

Furthermore, stemming and alphabetical ordering can be performed when extracting *n*-grams to filter out morphological, syntactical and semantic differences between linguistic expressions [CaMS01]. For example, the phrases "*information retrieval*", "*retrieve information*", and "*the retrieval of information*" are represented as the same bigram "*information retrieve*" after stemming and alphabetical ordering are performed. In our present work, we do not consider the influence of stemming and alphabetical ordering in *n*-gram generation, but leave it for future study.

Previous research has shown that the inclusion of bigrams tends to significantly improve performance. While the inclusion of longer *n*-grams, between *3* and *5*, may still benefit performance to some extent, the results on using them are inconclusive [Furn98, MlGr98]. Moreover, the inclusion of additional and longer *n*-grams introduces additional complexity and redundancy. As a compromise, we use *n*-grams of length up to *3*, and leave the investigation of longer *n*-grams to a future study.

4.1.3 Text Chunking

Chunking means breaking a sentence into a sequence of syntactically connected words, such as *noun* phrases and *verb* phrases. We use *YamCha* [KuMa00], a Support Vector Machine classifier, to perform chunk labeling, since it is the (freely available) system that performed the best in the *Computational Natural Language Learning* (CoNLL-2000) shared task, *Chunking and BaseNP chunking* task (*F-measure* 94% on a test corpus of 47,377 tokens). *YamCha* sequentially reads a set of tokens with POS tags, and makes the classification decision for each token. The feature set consists of the token itself, its POS tag, and its surrounding context which can be customized by the user.

Table 4.1.2 shows an example of text chunking by *Yamcha*. The chunk tag of the target token is determined by the token itself, the two preceding and two following tokens and their POS tags, as well as the dynamically predicted chunk tags for the two preceding tokens by the Support Vector Machine classifier.

Table 4.1.2. An example of text chunking by YamCha.

ICG-001	NN[1]	B-NP[2]
selectively	RB	B-ADVP
blocked	VBD	B-VP
the	DT	B-NP
beta-catenin-CBP	**NN**	**I-NP**
interaction	NN	I-NP
without	IN	B-PP
interfering	VBG	B-VP
with	IN	B-PP
the	DT	B-NP
beta-catenin-p300	NN	I-NP
interaction	NN	I-NP
.	.	O

4.1.4 Term Definition along Each Dimension

With the above preprocessing steps, we can generate a set of terms consisting of single words, statistical phrases, or syntactical phrases from raw text. Since we try to classify a fragment along five dimensions with different characteristics, the formation of terms should vary with the dimension such that each dimension has its own term set consisting

[1] Yamcha uses the Penn Treebank [MaSM94] POS tag set.

[2] B-CHUNK stands for the first word of the chunk, I-CHUNK stands for every other word inside the chunk, and O chunk is used for tokens which are not part of any chunk.

of the most distinguishing terms. We next analyze in detail the choice of terms along each dimension.

Focus

From the analysis of the manually annotated data, we found that we can typically determine the *Focus* of a sentence based on the presence of certain words or phrases without analyzing its syntactical structure. As examples[3], we consider the following sentences,

> *We determined* TP53 gene mutation in two cases and the genome-wide allelotype, AXIN1, and CTNNB1/beta-catenin gene mutation in one case. ***1SP3E3*
>
> The structural features of substrate recognition by calpains *are not yet fully understood*. ***1GN3E0*
>
> DNA sequence *was collected and analyzed* on an ABI Prism 377 automated DNA sequencer. ***1MP3E3*

In the first sentence, the statistical phrase "*We determined*" implies that a certain finding by the authors follows. Other biomedical words or phrases such as "*TP53*", "*gene mutation*", "*genome-wide allelotype*", "*AXIN1*", and "*CTNNB1/beta-catenin*" further verify that the topic of the sentence is a specific biological finding or experimental result. In the second sentence, the phrase "*are not yet fully understood*" describes the current status of a certain phenomenon, which makes the sentence a generic statement. In the

[3] All the annotated examples in this Thesis are cited from the Annotation Guidelines [ShWR06].

third sentence, the phrase "*was collected and analyzed*" introduces a certain methodology adopted by the authors. The above examples show that the *Focus* of a fragment is mainly determined by the presence or absence of words or phrases. Hence, we define the terms for the *Focus* dimension as single words and statistical phrases.

Polarity

To determine whether an assertion is stated positively or negatively, a shallow analysis of the sentence structure is helpful. As examples, consider the following sentences:

> Epidemiological data *do not* support the link between MMR vaccination and the development of autism [13]. ***1SN3E2*
>
> Aberrant overproduction of soluble Wnt antagonists by MM cells in the BM microenvironment may therefore impair *not only* bone formation but also normal processes of hematopoiesis. ***1SP1E0-*

Although they both contain the word "*not*", the verb phrase "*do not*" in the first sentence indicates that the sentence is stated negatively. In contrast, the adverb phrase "*not only*" in the second sentence merely specifies the range of the object, and the statement is expressed in the affirmative despite the presence of the word "*not*".

We perform shallow syntactical analysis, including POS tagging and text chunking, on the sentences, and use single words and syntactical phrases as terms for the *Polarity* dimension.

Certainty

After analyzing the manual annotations, we found that the *Certainty* level of a fragment can often be decided by the simple statistics of words, such as the presence or absence of certain words or phrases. For instance, cue words such as *indicate*, *suggest*, *demonstrate*, *determine*, *may*, *perhaps*, and cue phrases such as *very likely*, *it is unknown if*, or *it is unclear whether* can help decide the *Certainty* of an assertion. Therefore, we define the terms for the *Certainty* dimension as single words and statistical phrases. We use the following two examples to illustrate the importance of incorporating statistical phrases in the term set:

> Doctors look for variations that consistently *appear* in the DNA of family members with the disorder. ***1MP3E0*
>
> Thus, *it does not appear that* our findings concerning circadian rhythms were the result of the children being lower functioning as reported in previous investigations (Jensen et al., 1985). ***1SN1E23*

The single word "*appear*" in the first example does not carry much information about the certainty level of the sentence; while the phrase "*it does not appear that*" in the second example indicates that the certainty level of the sentence is *Low certainty*. Hence, if the data representation only consists of single words, such distinctive information related to the classification will be lost when phrases are not retained.

Evidence

Similar to the *Focus* and *Certainty* dimensions, we have found from the training data that the presence of certain words or statistical phrases can typically suggest the *Evidence* level of a fragment. As an example, consider the fragment,

> In Cblm, Purkinje cells stained as early as 2 wk; ***1SP3E3*

The past tense word "*stained*" indicates an experimental result, thereby the *Evidence* level of the fragment is *E3*, i.e. *Explicit evidence*. The contribution of statistical phrases to the decision on the *Evidence* level can be illustrated through the following two examples:

> Because a canonical Wingless-type (Wnt) signaling pathway *has recently been shown* to play an important role in osteoblast differentiation, ***1SP3E1*

> *We show that* a simple one-step procedure using CD3-magnetic beads to render the malignant T cells apoptotic and the separation column matrix to simultaneously activate monocytes results in overnight production of apoptotic cell-loaded DC. ***1MSP3E3+*

In the first example, the phrase "*has recently been shown*" suggests that the study was previously performed by others, and the evidence exists but is not explicitly specified here. Hence, the *Evidence* level is *E1*, formally, *Claim of evidence without verification*. In the second example, the phrase "*We show that*" states that the evidence has been explicitly provided somewhere within the paper, and accordingly, the *Evidence* level is

E3, that is, *Explicit evidence*. These two examples illustrate that the distinguishing information carried by statistical phrases cannot be substituted by single words or syntactical phrases. Therefore, we define the terms for the *Evidence* dimension as single words and statistical phrases.

Trend

The *Trend* dimension is defined to describe the positive or negative direction of certain phenomenon from the semantic perspective. Consider the following sentence as an example,

> In fact, as demonstrated using several SOD assays including pulse radiolysis, 2-ME does not *inhibit* SOD ***1SN3E3-*
> but rather *interferes* with the SOD assay originally used. ***2SP3E3-*

Because the semantic meaning of the words "*inhibit*" and "*interferes*" are negative, the *Trend* of both fragments is *Decrease,* although from the structural perspective, the first fragment is stated affirmatively and the second one is stated negatively.

In this case, we use single words as well as syntactical phrases to form the terms, as we believe that the shallow syntactical analysis of a fragment can help to better decode its semantic meaning.

After transforming the raw text into a set of terms, we next need to reduce the high dimensionality of the term space by removing unnecessary or noisy terms. As previously mentioned in Section 2.2.1, in text categorization, since the number of training examples is typically small compared to the large number of terms, it is important to reduce the size of the term set to avoid overfitting and achieve a good classification performance.

4.2 Term Space Reduction

In Section 2.2.1 we have briefly introduced several approaches for dimension reduction, such as stemming, *stop word* removal, term selection based on term space reduction (TSR) functions, and term extraction based on term clustering or other techniques (e.g. Principal Component Analysis or Latent Semantic Indexing). In addition to these methods, terms can be selected according to their syntactical roles in our classification tasks. We next discuss in detail the methods we adopt for term space reduction.

4.2.1 Stemming

As discussed in Section 2.2.1, stemming means removing common morphological suffixes from words. We employ the Porter Stemmer [Port80] to perform stemming. The algorithm consists of several steps, and at each step the suffix of a word is stripped according to a set of fixed rules, for instance, replacing the suffix "*tional*" with "*tion*", and the suffix "*tion*" with "*t*". Complex suffixes are removed step by step. For example, "*congratulations*" is stripped to "*congratulation*", then to "*congratulate*", then to "*congratul*". We apply stemming for the *Polarity* and *Trend* dimensions, since the suffix-removal does not change the *Polarity* or the *Trend* of a word. We do not consider stemming for the other three dimensions, since it may degrade the classification performance. For example, when the authors try to introduce their methods and experimental results, they typically use the past tense. Therefore, such morphological forms can be important in determining the *Focus* or the *Evidence* level of a fragment.

4.2.2 Stop Word Removal

Stop words typically refer to frequent but uninformative words, such as *articles*, *pronouns*, and *prepositions*. Considering the nature of our multi-dimensional classification task, we define a different set of *stop words* for each dimension (Refer to Appendix E for the details). For instance, *pronouns* such as *we* or *their* are defined as *stop words* for the *Trend* dimension, but not for the *Evidence* dimension. This is because in the context of *Evidence*, they are important in distinguishing whether the work is done by the authors themselves or by others.

4.2.3 Removal of Terms by Part-of-Speech Tags

From the annotated examples we learned that the syntactical category of a word might decide its relative importance for the classification along different dimensions. For example, *nouns*, *pronouns*, and *prepositions* do not convey much information regarding the *Polarity* of a fragment. *Prepositions* and *nouns* are less important for the determination of the *Certainty* level than *verbs*, *adjectives* and *adverbs*. *Pronouns* and *prepositions* typically convey little information about the *Trend* of a fragment. Therefore, terms can be selected based on the part-of-speech (POS) tags associated with them. Table 4.2.1 defines the criteria of term selection based on POS tags for each classification dimension, where 1 indicates the inclusion of terms associated with the POS tag, and 0 indicates the exclusion of terms associated with the POS tag. We present here only the preliminary criteria, which need to be further revised based on future experiments.

Table 4.2.1. Term selection based on part-of-speech (POS) tags. F denotes Focus, P denotes Polarity, C denotes Certainty, E denotes Evidence, and T denotes Trend. Words in parentheses are examples from the corresponding syntactical categories. Words in italics indicate that the corresponding syntactical categories are specifically defined for the words.

POS	F	P	C	E	T	Syntactical Category	POS	F	P	C	E	T	Syntactical Category
CC	1	1	0	1	1	coordinating conjunction(but)	VBN	1	1	1	1	1	participle *been*
CS	0	0	0	1	0	subordinating conjunction	VBZ	1	1	1	1	1	3rd person singular *is*
CSN	0	0	0	0	0	comparative conjunction (than)	VDB	1	1	1	1	1	base *do*
CST	0	0	1	1	0	complementizer (that)	VDD	1	1	1	1	1	past *did*
DB	1	1	1	1	1	predeterminer (such)	VDG	1	1	1	1	1	participle *doing*
DD	0	0	0	0	0	determiner (the)	VDI	1	1	1	1	1	infinite *do*
EX	1	0	1	1	0	existential *there*	VDN	1	1	1	1	1	participle *done*
GE	0	0	0	1	0	genitive marker *’s*	VDZ	1	1	1	1	1	3rd person singular *does*
II	1	1	0	1	1	preposition	VHB	1	1	1	1	1	base *have*
JJ	1	1	1	1	1	adjective	VHD	1	1	1	1	1	past *had*
JJR	1	1	1	1	1	comparative adjective	VHG	1	1	1	1	1	participle *having*
JJT	0	0	1	1	0	superlative adjective	VHI	1	1	1	1	1	infinitive *have*
MC	1	1	1	1	1	number or numeric	VHN	1	1	1	1	1	participle *had*
NN	1	0	0	1	0	noun	VHZ	1	1	1	1	1	3rd person singular *has*
NNP	1	0	0	1	0	proper noun	VVB	1	1	1	1	1	base form lexical verb
NNS	1	0	0	1	0	plural noun	VVD	1	1	1	1	1	past tense lexical verb
PN	1	0	1	1	0	pronoun (we)	VVG	1	1	1	1	1	present participle
PND	0	0	1	1	0	determiner as pronoun (this)	VVI	1	1	1	1	1	infinitive lexical verb
PNG	0	0	1	1	0	genitive pronoun (our)	VVN	1	1	1	1	1	past participle
PNR	0	0	0	0	0	relative pronoun (which)	VVZ	1	1	1	1	1	3rd person singular
RR	1	1	1	1	1	adverb	VVNJ	1	1	1	1	1	prenominal past participle
RRR	1	1	1	1	1	comparative adverb	VVGJ	1	1	1	1	1	prenominal present participle
RRT	0	0	1	1	0	superlative adverb	VVGN	1	1	1	1	1	nominal gerund
SYM	1	0	0	0	0	symbol	(	0	0	0	0	0	left parenthesis
TO	0	0	0	0	0	infinitive marker *to*	)	0	0	0	0	0	right parenthesis
VM	1	1	1	1	1	modal (can)	,	0	0	0	0	0	comma
VBB	1	1	1	1	1	base *be, am, are*	.	0	0	0	0	0	end-of-sentence period
VBD	1	1	1	1	1	past *was, were*	:	0	0	0	0	0	dashes, colons
VBG	1	1	1	1	1	participle *being*	“	0	0	0	0	0	left quote
VBI	1	1	1	1	1	infinitive *be*	“	0	0	0	0	0	right quote

4.2.4 Removal of Terms by Phrase Type

Similar to the term selection based on POS tags, syntactical phrases can be further selected according to their syntactical role. As we discussed in Section 4.2.1, the term sets for the *Polarity* and *Trend* dimensions consist of single words and syntactical phrases. We define the criteria for selecting syntactical phrases in Table 4.2.2, where 1 indicates the inclusion of the phrase type, and 0 indicates its exclusion. We present here only the preliminary criteria, which may be further revised based on future experiments.

Table 4.2.2. Term selection based on syntactical phrase type. Words in parentheses are examples from the corresponding syntactical categories.

Type	Polarity	Trend	Syntactical Category
ADJP	1	1	Adjective Phrase
ADVP	1	1	Adverb Phrase
CONJP	1	1	Conjunction Phrase (as well as)
INTJ	0	0	Interjection (wow)
LST	0	0	List marker (1. 2. 3.)
NP	0	1	Noun Phrase
PP	0	0	Prepositional Phrase
PRT	0	1	Particle (look *up*, slow *down*)
SBAR	0	0	Relative clauses and Subordinate clauses (who, what)
VP	1	1	Verb Phrase

4.2.5 Removal of Nouns Specific to the Biomedical Domain

We learned from the annotated examples that biomedical concepts or terms, such as gene or protein names, do not convey much information about the *Evidence* of a fragment. Therefore, nouns specific to the biomedical domain should be removed from the final term set for the *Evidence* dimension. We do not consider here the *Certainty* and *Polarity* dimensions, since nouns in general are removed from the final term sets for these two dimensions as we discussed above. We can identify gene or protein names using existing tools, for example, a gene and protein name tagger developed by Tanabe and Wilbur [TaWi02]. Currently, the removal of nouns specific to the biomedical domain is not implemented in this work, and we will leave it for future studies.

4.2.6 Advanced Term Selection and Term Extraction Methods

To obtain the set of most distinguishing terms that yield the best classification performance, we can further examine several term selection functions surveyed in Section 2.2.1, such as *chi-square, information gain*, and *Z score*. These functions measure how predictive a term is for certain categories. We can select terms based on their scores measured by these functions, and investigate the classification performance when various numbers of terms are selected. However, there are two major drawbacks for dimension reduction by term selection functions. First, the selected term set may contain redundancy. Since highly correlated terms tend to score similarly, it is possible that correlated terms are chosen together as the most distinguishing terms. Second, the contributions of a large number of terms scoring relatively low are completely ignored. As discussed by Sebastiani [Seba99], it is quite possible that the combination of the terms, each with a small amount of critical information, can produce a high-order distinguishing feature. To address these issues, we further examine the classification performance with term extraction methods, namely, Principal Component Analysis (PCA) or Latent Semantic Indexing (LSI).

We summarize the preprocessing steps for the classification along the five dimensions in Figures 4.2.1 and 4.2.2.

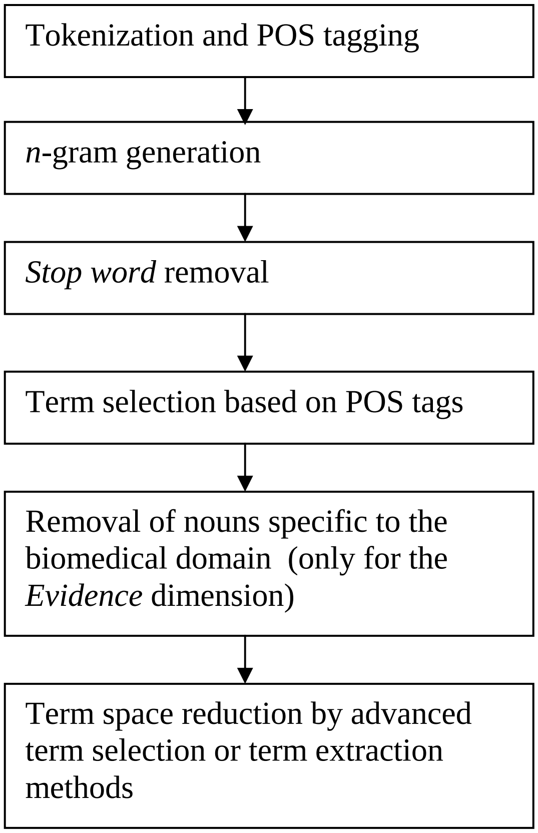

Tokenization and POS tagging
n-gram generation
Stop word removal
Term selection based on POS tags
Removal of nouns specific to the biomedical domain (only for the *Evidence* dimension)
Term space reduction by advanced term selection or term extraction methods

Figure 4.2.1. Text preprocessing for the Focus, Certainty, and Evidence dimensions.

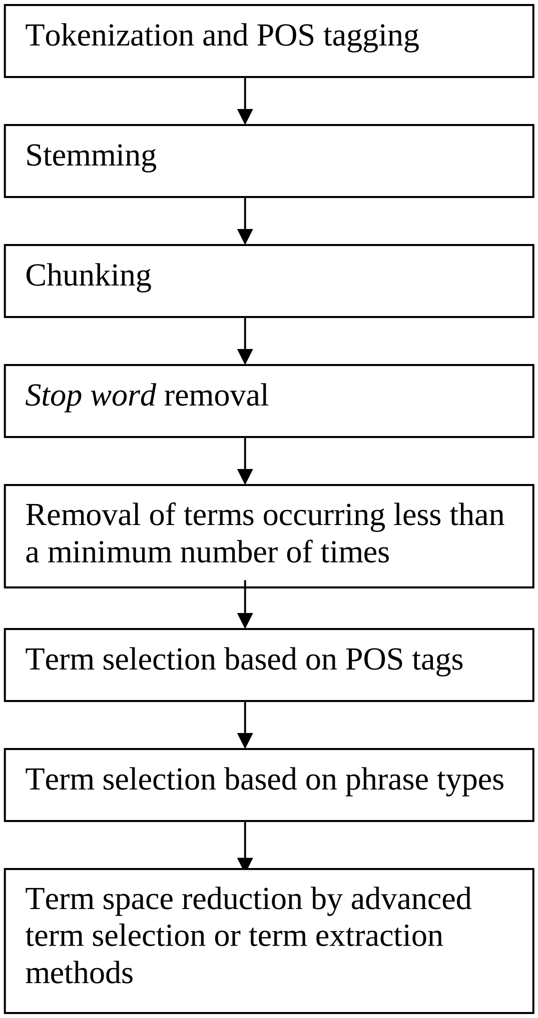

Tokenization and POS tagging
Stemming
Chunking
Stop word removal
Removal of terms occurring less than a minimum number of times
Term selection based on POS tags
Term selection based on phrase types
Term space reduction by advanced term selection or term extraction methods

Figure 4.2.2. Text preprocessing for the Polarity and Trend dimensions.

Following the above text preprocessing procedures, each fragment is represented as a fixed-length vector of terms. For the *Focus*, *Certainty*, and *Evidence* dimensions, terms are defined as single words and statistical phrases. For the *Polarity* and *Trend* dimensions, terms are defined as single words and syntactical phrases. As to the choice of term weighting scheme, we start from the simplest approach, *binary weighting*, with 1 representing the presence of a term, and 0 representing its absence. In future work, we will examine other weighting schemes, such as the widely-used $TF \cdot IDF$ weighting scheme, and the $TF \times RF$ [4]scheme proposed by Lan *et al.* [LTLS05], to further improve the discriminating power of individual terms.

[4] $RF = \log \frac{1+n_i}{\overline{n}_i}$, where n_i is the number of fragments that contain the term t_i, and $\overline{n}_i$ is the number of fragments that do not contain the term t_i.

Chapter 5

Classification Methodology

In this chapter, we first analyze the novelty and challenge in the multi-dimensional fragment classification. After briefly discussing the choice of classification algorithms for each dimension, we introduce our Maximum Entropy model that is designed to address the special issues of our classification task. We start from the theory of Maximum Entropy, then provide a thorough discussion of the model developed here.

5.1 Challenges in Fragment Classification

The novelty of our classification scheme lies in the fact that it tries to describe the same object from several different perspectives. More specifically, it tries to classify the same fragment along five dimensions, in contrast to existing methods that typically classify the data along one dimension. The fragment classification has several distinctive characteristics, most notably along the *Focus* and *Evidence* dimensions:

First, both *Focus* and *Evidence* are multi-label classifications. The *Focus* of a fragment can be one of the categories: *Scientific*, *Generic*, *Methodology*, or their combination. Consider the following sentence as an example:

> Future structural and functional studies will be necessary to understand precisely how She2p binds *ASH1* mRNA and how interactions with She3p influence the formation of a functional localization complex. ***1SGP0E0*

The sentence poses scientific questions: “*how She2p binds ASH1 mRNA*” and “*how interactions with She3p influence the formation of a functional localization complex*”. Moreover, it talks about the general trend of knowledge, “*Future structural and functional studies will be necessary*”. Therefore, the *Focus* of the sentence is *SG*, that is, both *Scientific* and *Generic*.

Similarly, the classification along the *Evidence* dimension is also multi-labeled, for example, the fragment:

> …the overexpression of phospho-H2Av did not induce G2/M arrest or affect DSB-dependent G2/M arrest (fig. S10) (14,21), ***1SN3E23+*

contains a reference to an experimental figure “*fig. S10*” within the paper and citations of other papers “*(14,21)*”. In such a case, the evidence level of the fragment is *E23*, that is, *Explicit citation* (denoted as *E2*) and *Explicit evidence* (denoted as *E3*).

Second, the classification along the *Focus* dimension is context dependent. The topic of a fragment may be determined not only by the fragment itself, but also by the whole sentence. For example,

> The children with autism, ***1SP3E0*
> but not typical children, ***2SN3E0*
> showed a more variable circadian rhythm as well as statistically significant elevations in cortisol following exposure to a novel, nonsocial stimulus. ***3SP3E3+*

The contents of the first two fragments alone concern general knowledge rather than detailed scientific findings. However, the third fragment turns the main topic of the whole sentence into a specific scientific fact, i.e. the characteristics of children with autism compared to normal children. Considering the context of the whole sentence, the *Focus* of both of the first two fragments is *S*, that is, *Scientific*.

Third, the classification along the *Focus* and *Evidence* dimensions may be correlated. For example, when a fragment discusses a certain methodology, it usually describes scientific experiments that have been done, and consequently the *Evidence* level is *Explicit evidence*. Consider the following sentence as an example:

> Mononuclear cells (MNC) were isolated by centrifugation over a ficollhypaque gradient followed by two washes in RPMI 1640 (Gibco, Gaithersburg, MD) containing 10% AB serum and 2mM EDTA. ***1MP3E3*

It is hard to decide that the *Evidence* level of the sentence is *Explicit evidence* solely based on the presence or absence of the cue terms (e.g. *Fig.*, *Table*, Citation, cue phrases such as *we found that*, *our results show*). However, it is easier to detect that the *Focus* of the sentence is an experimental method. Because different term sets are used for different dimensions, terms predictive for the *Focus* dimension such as "*MNC*", "*RPMI*", "*was isolated*", are either considered less important or filtered out for the classification along the *Evidence* dimension (nouns specific to the biomedical domain are removed for the *Evidence* dimension as discussed in Section 4.2.5). In such a case, it is helpful if we take into account the connection between the *Focus* category *Methodology* and the *Evidence* category *Explicit evidence* when performing classification. Hence, to improve the performance, the classification should be done such that the decisions along these two dimensions can mutually influence each other.

To address special aspects of our classification challenge, a new classification model based on Maximum Entropy is designed especially for the fragment classification along the *Focus* and *Evidence* dimensions. We discuss the design of the model in detail in Section 5.3.

The classification along the other three dimensions, *Polarity*, *Trend*, and *Certainty*, can be treated as individual text classification tasks and can be performed separately. Since the basic classification unit is a sentence fragment with a few words, the data representation is relatively sparse. We choose Support Vector Machines (SVMs) for the classification along the *Polarity* and *Trend* dimensions because, as discussed in Section 2.2.2, SVMs work well on sparse data and they usually outperform other algorithms in text classification tasks. As for the classification along the *Certainty* dimension, we examine ranking classifiers such as SVM and Naïve Bayes, so that we can better learn the uncertainty of the classification decisions over the categories.

We have briefly introduced the underlying theory of Naïve Bayes and SVM in Section 2.2.2. In this chapter, we focus on the Maximum Entropy model specifically designed to address the three problems of the fragment classification along the *Focus* and *Evidence* dimensions.

5.2 Theory of Maximum Entropy

As previously stated in Section 2.2.2, the underlying principle of Maximum Entropy is to model everything that is known, and assume nothing that is unknown, i.e. choose a probability distribution that will satisfy any known constraints, while otherwise being as uninformative as possible. For instance, in a text classification problem, we typically want to classify documents into two categories, *relevant* or *irrelevant* with respect to a given topic. Suppose we learned from the training data that 70% of the documents that contain the word *mice* are relevant documents. Then we can build a classification model based on the following rule: if a document contains the word *mice*, the probability that it belongs to the *relevant* category is 70%, while the probability that it belongs to the *irrelevant* category is 30%; otherwise, the probability distribution for the two categories is uniform, 50% each. This model is a simple Maximum Entropy model. It is consistent with the known constraints, and makes no assumptions about what is unknown.

In general, Maximum Entropy can be used to estimate any probability distribution. Since we are only interested in text classification, we limit our discussion to conditional probability distributions. Specifically, we estimate the conditional probability $p(c \mid d)$, where d is a data example, and c is the possible category label.

In this section, we briefly introduce the theory of Maximum Entropy. Detailed information can be found in Appendix A, as well as in several papers [BeDD96, Berg97, NiLM99, Berg00, Taka04]. Here we interpret the probability $p(c \mid d)$ on an event space of documents, that is, d represents a document.

5.2.1 Features and Constraints

In text classification, suppose we denote a set of training documents D associated with labels C as $Observed(D,C)$. With Maximum Entropy, our goal is to construct a classification model, specifically, a conditional probability distribution, $p(c \mid d)$, which could have generated the given set of training data, $Observed(D,C)$. In other words, this model distribution, $p(c \mid d)$, will reflect a set of statistical facts derived from the training data. To do this, we first describe any useful statistic as a real-valued function, f , which is called a *feature function* or *feature*. We then constrain the expected value of f calculated from the model distribution, $p(c \mid d)$, to be the same as that derived from the training data, $Observed(D,C)$. We can use any real-valued function of a document-label pair $(d,c) \in (D,C)$ as a feature. The feature functions typically express some useful statistics of documents and categories, for instance, a co-occurrence relation between terms (words or phrases) and categories. We give here an example of the definition of a feature function.

Suppose there are a total of *m* categories, we can use an *m*-dimensional vector, $c \equiv (c_1,...,c_m)$, to denote any possible label that can be assigned to a document, defined as:

$$c_i = \begin{cases} 1 \;\; \textit{if the document is assigned to category i}; \\ 0 \;\textit{otherwise.} \end{cases}$$

As surveyed in Section 2.2.1, each document can be represented as a set of terms. Suppose that the term space consists of *k* terms, we use a *k*-dimensional vector to represent a document, $d \equiv (t_1^d,...,t_k^d)$, defined as:

$$t_j^d = \begin{cases} 1 \;\; \textit{if term j is present in document d} \; ; \\ 0 \;\textit{otherwise.} \end{cases}$$

To capture the co-occurrence between terms and categories for a given document-label pair (d,c), we define a matrix $F(d,c)$ as the product of the category vector $(c_1,...,c_m)'$ (the transpose vector of *c*) and the term vector $(t_1^d,...,t_k^d)$:

$$F(d,c) = c'd = (c_1,...,c_m)'(t_1^d,...,t_k^d)\,.$$

$F(d,c)$ is an $m \times k$ matrix. The rows correspond to categories, and the columns correspond to terms. The Boolean element $F_{ij}(d,c)$ represents whether category i and term j co-occur. From the definition of the matrix $F(d,c)$, we know that $F_{ij}(d,c)$ is 1 when both c_i and t_j^d are 1, formally:

$$F_{ij}(d,c) = \begin{cases} 1 \ \ \mathit{if\ term\ j\ is\ present\ in\ document\ d\ and\ document\ d\ is\ assigned\ to\ category\ i}; \\ 0 \ \mathit{otherwise}. \end{cases}$$

We refer to the matrix $F(d,c)$ as a *feature function matrix* or *feature matrix*, where each element $F_{ij}(d,c)$ corresponds to a feature function between a document-label pair (d,c). We can also transform the matrix $F(d,c)$ into a vector, denoted as $f(d,c)$:

$$f(d,c) = (F_{11}(d,c),...,F_{1k}(d,c),F_{21}(d,c),...,F_{2k}(d,c),...,F_{m1}(d,c),...,F_{mk}(d,c)).$$

Let $n = m \times k$, each element of the vector $f(d,c)$ is denoted as a feature $f_i(d,c)$, where $i \in [1,2,...,n]$. Thus, for a given document-label pair (d,c), we have defined a total of $m \times k$ features.

If we denote the observed probability (empirical probability) that a document-label pair (*d*, *c*) occurs in the training examples as $\tilde{p}(d,c)$, the expected value of the feature $f_i(d,c)$ for the training examples can be calculated by:

$$E_{\tilde{p}}(f_i) = \sum_{d,c \in Observed(D,C)} \tilde{p}(d,c) f_i(d,c). \tag{5.1}$$

The expected value of $f_i(d,c)$ with respect to a given distribution $p(c \mid d)$ can be calculated by:

$$E_p(f_i) = \sum_{d \in D} p(d) \sum_{c \in C} p(c \mid d) f_i(d,c), \tag{5.2}$$

where $p(d)$ is the probability of a document d to occur in the corpus. In practice, the probability is unknown and we have no interest in modeling it. Therefore, we use the occurrence frequency of a document in the training corpus, $\tilde{p}(d)$, as an approximation:

$$E_p(f_i) = \sum_{d \in D} p(d) \sum_{c \in C} p(c \mid d) f_i(d,c) \approx \sum_{d \in D} \tilde{p}(d) \sum_{c \in C} p(c \mid d) f_i(d,c) . \tag{5.3}$$

Each document is considered to occur only once[1] in the training corpus, therefore, $\tilde{p}(d) = \frac{1}{|D|}$, where $|D|$ is the total number of training documents. We then restrict the model distribution $p(c \mid d)$ to have the expected value for each feature $f_i(d,c)$ as derived from the training data:

$$E_{\tilde{p}}(f_i) = E_p(f_i) \ , \tag{5.4}$$

$$\text{i.e.} \quad \sum_{(d,c) \in Oberved\,(D,C)} \tilde{p}(d,c) f_i(d,c) = \sum_{d \in D} \tilde{p}(d) \sum_{c \in C} p(c \mid d) f_i(d,c) . \tag{5.5}$$

In summary, there are two main steps in defining a Maximum Entropy model for text classification: first, define a set of features; second, calculate the expected value of each feature from the training data, and impose this as a constraint on the model.

5.2.2 Maximum Entropy Principle

The principle of Maximum Entropy is to find the least informative (the most uncertain) model that also satisfies any given constraints. A mathematical measure of the uncertainty of a conditional probability distribution $p(c \mid d)$ is provided by the conditional entropy [CoTh91]:

[1] If the same document representation occurs multiple times in the training set, each time it is considered to represent a different document.

$$H(p) \equiv -\sum_{d \in D} \sum_{c \in C} p(d) p(c \mid d) \log p(c \mid d) \tag{5.6}$$

$$\approx -\sum_{d \in D} \sum_{c \in C} \widetilde{p}(d) p(c \mid d) \log p(c \mid d) \,. \tag{5.7}$$

The problem of Maximum Entropy is to find a distribution $p^*(c \mid d)$ that has the maximum entropy value $H(p^*)$ among all the conditional probability distributions $p(c \mid d)$ satisfying the given constraints:

$$\sum_{(d,c) \in Oberved\ (D,C)} \widetilde{p}(d,c) f_i(d,c) = \sum_{d \in D} \widetilde{p}(d) \sum_{c \in C} p(c \mid d) f_i(d,c) \,.$$

When the constraints are defined in this fashion, it is guaranteed that a unique distribution $p^*(c \mid d)$ exists, which has the maximum entropy among all the distributions $p(c \mid d)$ satisfying the constraints (See Appendix A for details). Moreover, the distribution $p^*(c \mid d)$ is of a parametric exponential form:

$$p^*(c \mid d) = \frac{\exp\{\sum_i \lambda_i f_i(d,c)\}}{Z(d)} \,, \tag{5.8}$$

where λ_i is a parameter to be estimated, and *Z(d)* is the normalizing factor that ensures the sum of the probability $p^*(c \mid d)$ with respect to each $c \in C$ equals to one, defined as:

$$Z(d) = \sum_{c \in C} \exp\{\sum_i \lambda_i f_i(d,c)\} \,. \tag{5.9}$$

5.2.3 Maximum Entropy and Maximum Likelihood

Given a set of training examples, it was proven that the solution to the Maximum Entropy problem is also the solution to the Maximum Likelihood problem for the probability distribution with the same exponential form [BeDD96] (See Appendix A for details).

That is, the constrained maximization of entropy is equivalent to the unconstrained maximization of the likelihood of a set of exponential distributions. The model distribution with the maximum entropy is also the one, among all the models of the same parametric form, that fits the training examples best.

The problem of Maximum Entropy is thus converted into the problem of maximizing the likelihood of a set of exponential models with respect to the training data. Given the training dataset, $Observed(D,C)$, the log likelihood of the model distribution $p(c \mid d)$ is defined as follows [BeDD96]:

$$L(p) \equiv \log \prod_{(d,c) \in Observed(D,C)} p(c \mid d)^{\tilde{p}(d,c)} . \tag{5.10}$$

The joint probability, $\tilde{p}(d,c)$, measures the frequency that a particular document-label pair (d,c) occurs in the training set. As previously stated, we consider a non-redundant dataset in which each document only occurs once. Therefore, the joint probability, $\tilde{p}(d,c)$, is estimated by:

$$\tilde{p}(d,c) = \begin{cases} \frac{1}{|D|} & if \;\; (d,c) \in Observed\,(D,C); \\ 0 & otherwise, \end{cases} \tag{5.11}$$

where $|D|$ is the total number of training examples. In practice, instead of using the empirical probability $\tilde{p}(d,c)$, we can use the number of times that a particular pair (d,c) occurs in the training set, (i.e. 0 or 1), to calculate the log likelihood of the model distribution $p(c \mid d)$:

$$L(p) = \log \prod_{(d,c) \in Observed(D,C)} p(c \mid d) . \tag{5.12}$$

Substituting $p(c \mid d) = \frac{\exp\{\sum_i \lambda_i f_i(d,c)\}}{Z(d)}$ into the above equation, we can get the log likelihood of the model parameters $\Lambda \equiv (\lambda_1, \lambda_2, ...\lambda_n)$:

$$L(\Lambda) = \sum_{(d,c)\in Observed(D,C)} \left(\sum_i \lambda_i f_i(d,c) - \log \sum_{c\in C} \exp\{\sum_i \lambda_i f_i(d,c)\}\right). \quad (5.13)$$

To find the parameters Λ^* that maximize the log likelihood $L(\Lambda)$, we need to resort to numerical methods. One method specifically designed for the Maximum Entropy problem is the Improved Iterative Scaling (IIS) algorithm [Berg97]. We give a brief introduction of IIS in the following section.

5.2.4 Parameter Estimation -- Improved Iterative Scaling (IIS)

Since the log likelihood function, $L(\Lambda)$, is concave[2] with respect to each parameter λ_i, it is guaranteed to have a single global maximum. Starting from an initial set of parameters $\Lambda^0 \equiv (\lambda_1^0, \lambda_2^0, ...\lambda_n^0)$, IIS will find an incrementally more likely set of parameters at each iteration until the log likelihood, $L(\Lambda)$, converges to the global maximum. This can be done by ensuring the new model has a higher likelihood at each step, that is,

$$L(\Lambda + \delta) - L(\Lambda) > 0, \quad (5.14)$$

where $\delta \equiv (\delta_1, \delta_2, ..., \delta_n)$, and δ_i is the change in the parameter λ_i at each step. Formally, $\delta_i = \lambda_i^t - \lambda_i^{t-1}$, where *t* is the iteration number. Substituting equation (5.13) into inequality (5.14), we get:

[2] A function $f(x)$ is concave on an interval [a,b] if for any two points x_1 and x_2 in [a,b] and any ∂ where $0 < \partial < 1$, $f(\partial x_1 + (1-\partial)x_2) \geq \partial f(x_1) + (1-\partial) f(x_2)$.

$$L(\Lambda+\delta)-L(\Lambda)=\sum_{(d,c)\in Observed\ (D,C)}\Big(\sum_i \delta_i f_i(d,c)-\log\frac{\sum_{c\in C}\exp\{\sum_i(\lambda_i+\delta_i)f_i(d,c)\}}{\sum_{c\in C}\exp\{\sum_i \lambda_i f_i(d,c)\}}\Big)>0\,\cdot$$

(5.15)

Using the inequality $-\log\partial\geq 1-\partial$ and Jensen's inequality[3], we can find a lower bound for inequality (5.15) (Refer to Appendix B for the derivation details):

$$L(\Lambda+\delta)-L(\Lambda)\geq\underbrace{\sum_{(d,c)\in Observed(D,C)}\Big(\sum_i\delta_i f_i(d,c)\ +\ 1\ -\ \sum_{c\in C}p(c\,|\,d)\sum_i\frac{f_i(d,c)}{f^{\Sigma}(d,c)}\exp\{\delta_i f^{\Sigma}(d,c)\}\Big)}_{A(\delta|\lambda)},$$

(5.16)

where $f^{\Sigma}(d,c)\equiv\sum_i f_i(d,c)$.

We denote the right hand side of inequality (5.16) as $A(\delta\,|\,\lambda)$. Since $A(\delta\,|\,\lambda)$ is a lower bound on the change in the log likelihood, $L(\Lambda+\delta)-L(\Lambda)$, we can guarantee an increase in the likelihood if $A(\delta\,|\,\lambda)$ is positive. To maximize the difference in the log likelihood, i.e., to best improve the model, we first solve for the maximum of $A(\delta\,|\,\lambda)$ with respect to $\delta\equiv(\delta_1,\delta_2,...,\delta_n)$. By restricting this maximum to be positive, we can find a change δ_i in each λ_i that will improve the model likelihood. The maximum of $A(\delta\,|\,\lambda)$ can be obtained by differentiating $A(\delta\,|\,\lambda)$ with respect to each δ_i and setting the derivatives to zero:

$$\frac{\partial A(\delta\,|\,\lambda)}{\partial\delta_i}=\sum_{(d,c)\in Observed(D,C)}\Big(f_i(d,c)-\sum_{c\in C}p(c\,|\,d)\,f_i(d,c)\exp\{\delta_i f^{\Sigma}(d,c)\}\Big)=0\,. \qquad (5.17)$$

We can use a root finding procedure (e.g. Newton's method) to solve equation (5.17) and obtain the optimal set of changes $\delta\equiv(\delta_1,\delta_2,...\delta_n)$ at each iteration. The

[3] If *f* is a convex function and *x* is a random variable then $E(f(x))\geq f(E(x))$.

process repeats until the likelihood converges to the global maximum. Table 5.2.1 summarizes the IIS algorithm.

Table 5.2.1. An outline of the IIS algorithm for parameter estimation.

```
IIS algorithm
        (1)    Start with an arbitrary value for each parameter $\lambda_i$
        (2)    Repeat until convergence:
               Set $\frac{\partial A}{\partial \delta_i}$=0 and solve for $\delta_i$
               Set $\lambda_i = \lambda_i + \delta_i$
```

To illustrate the principle of Maximum Entropy, we give an example of model construction in Appendix B. We refer readers to Appendix B for further information about how a Maximum Entropy classifier works in the area of text categorization.

5.3 Maximum Entropy Model for Multi-dimensional Fragment Classification

As discussed in Section 5.1, there are three distinctive characteristics for the fragment classification along the *Focus* and *Evidence* dimensions: multi-label classification, correlation between the two dimensions, and context dependent classification along the *Focus* dimension. In this section, we introduce, in detail, the Maximum Entropy model developed to address these issues. Specifically, the Maximum Entropy model is designed to classify fragments along the *Focus* and *Evidence* dimensions simultaneously, as well as to address the issues of multi-label and context dependent classification. We first introduce the categories and data representations, then focus on the feature definitions and the constraints that are imposed over the model.

fragments within it, we expect a more accurate prediction of the *Focus* of a sentence compared to that of a fragment. In the case when the *Focus* of a fragment is ambiguous for the classifier, we believe that a clear prediction for the *Focus* of the whole sentence, and the dependence of the *Focus* of a fragment on this context can help the classifier to make the right decision. As an example, consider the following sentence we have discussed in Section 5.1,

> The children with autism, ***1SP3E0*
> but not typical children, ***2SN3E0*
> showed a more variable circadian rhythm as well as statistically significant elevations in cortisol following exposure to a novel, nonsocial stimulus. ***3SP3E3+*

It is hard to predict the *Focus* of the first two fragments solely based on the few terms present in those fragments, while it is relatively easy to predict that the *Focus* of the whole sentence is *Scientific*. If the classification model can derive strong correlation between *Scientific Fragment Focus* and *Scientific Sentence Focus* from the training data, once we determine that the *Focus* of the sentence is *Scientific*, the final *Focus* classification of the first two fragments will tend to be biased towards *Scientific*.

Therefore, to improve the prediction accuracy for the *Focus* of a fragment, we introduce the third category space *Sentence Focus*. *Sentence Focus* concerns the *Focus* of a sentence, consisting of three categories: *Scientific*, *Generic*, and *Methodology*. We use a three-dimensional Boolean vector, $c_N \equiv (c_{N1}, c_{N2}, c_{N3})$, to represent the *Sentence Focus* label, defined as:

$$c_{Ni} = \begin{cases} 1 & if\ there\ is\ at\ lease\ one\ \ fragment\ \ whose\ Focus\ belongs\ to\ category\ i\ within\ the\ sentence; \\ 0 & otherwise. \end{cases}$$

Similar to the *Fragment Focus* label, there are seven possible values for the *Sentence Focus* label. We further discuss how to incorporate the correlation between *Fragment Focus* and *Sentence Focus* within the classification model in Section 5.3.3.

So far, we have defined three category spaces for the fragment classification along the *Focus* and *Evidence* dimensions: *Fragment Focus*, *Fragment Evidence*, and *Sentence Focus*. We illustrate the construction of the three category spaces by the following example,

> Although neuropathologic studies of autism are limited, ***1GP3E1*-
> reports of Purkinje and granule cell loss in Cblm (16) also suggest overlap
> with this neonatal infection paradigm. ***2SP2E2*

The value of *Fragment Focus* c_F is [**010]** for the first fragment, and [**100]** for the second fragment; the value of *Fragment Evidence* c_E is [**0100]** for the first fragment, and **[0010]** for the second fragment; the context of both fragments, i.e. the value of *Sentence Focus* c_N, is [**110]**.

5.3.2 Data Representation

As mentioned in Section 4.1, each fragment is represented as a vector of terms to be used as an input to classification algorithms. For each classification dimension, a different set of terms can be selected using term selection functions such as *chi-square* or *information gain* as discussed in Section 4.2. We use an *n*-dimensional Boolean vector, $t_F^d \equiv (t_{F1}^d,...,t_{Fn}^d)$, to represent a fragment for the *Focus* dimension, where *n* is the total number of terms selected for the *Focus* dimension. Each element t_{Fi}^d is defined by:

$$t_{Fi}^{d} = \begin{cases} 1 \ \ if \ \ term \ i \ is \ \ present \ in \ \ fragment \ d; \\ 0 \ otherwise. \end{cases}$$

Similarly, suppose a total of *m* terms are selected for the *Evidence* dimension, we can use an *m*-dimensional Boolean vector $t_{E}^{d} \equiv (t_{E1}^{d},...,t_{Em}^{d})$ to represent a fragment for the *Evidence* dimension. Each element t_{Ei}^{d} is defined by:

$$t_{Ei}^{d} = \begin{cases} 1 \ \ if \ \ term \ i \ is \ \ present \ in \ \ fragment \ d; \\ 0 \ \ otherwise. \end{cases}$$

To consider the surrounding context for the *Focus* of a fragment, i.e. the *Focus* of the sentence from which the fragment comes, we need a set of terms pertaining to the *Focus* of a sentence. Suppose that a total of *k* terms are selected to determine the *Focus* of a sentence. The context of a fragment can be represented by a *k*-dimensional Boolean vector, $t_{N}^{d} \equiv (t_{N1}^{d},...,t_{Nk}^{d})$, defined by:

$$t_{Ni}^{d} = \begin{cases} 1 \ \ if \ \ term \ i \ is \ \ present \ in \ the \ sentence \ where \ \ fragment \ d \ comes \ \ from; \\ 0 \ otherwise \ . \end{cases}$$

In summary, in the fragment classification along the *Focus* and *Evidence* dimensions, each fragment is associated with three types of labels: *Fragment Focus*, *Fragment Evidence*, and *Sentence Focus*. Accordingly, it is associated with three types of text representations: a vector of terms pertaining to the *Focus* of a fragment, a vector of terms pertaining to the *Evidence* of a fragment, and a vector of terms pertaining to the *Focus* of a sentence.

We use the concatenated Boolean vector, $d \equiv (t_{F}^{d}, t_{E}^{d}, t_{N}^{d})$, to denote a fragment, and the concatenated Boolean vector, $c \equiv (c_{F}, c_{E}, c_{N})$, to denote any possible label that can be assigned to a fragment. Since there are seven possible values for the *Fragment*

Focus label and the *Sentence Focus* label respectively, and five possible values for the *Fragment Evidence* label, there are a total of 245 (7*7*5) possible labels for each fragment. Our goal is to estimate the model distribution $p(c \mid d)$ given a set of training fragments *D* associated with labels *C*. To construct a Maximum Entropy classification model, we first need to define a set of features and constraints.

5.3.3 Features and Constraints

In the area of text categorization, a traditional Maximum-Entropy-based model typically imposes constraints over the following two statistical properties: the prior probability of each category, and the correlation between terms and categories. Accordingly, we introduce two types of feature function matrices: one captures the occurrence of categories within each category space, denoted as f_F^P, f_E^P, f_N^P respectively; the other captures the co-occurrence between terms and categories within each category space, denoted as f_F^T, f_E^T, f_N^T respectively. Suppose that c' denotes the transpose vector of *c*, for each fragment-label pair (d, c), the function matrices are defined as:

$$f_F^P(d,c) = c'_F\,, \tag{5.18}$$

$$f_E^P(d,c) = c'_E\,, \tag{5.19}$$

$$f_N^P(d,c) = c'_N\,, \tag{5.20}$$

$$f_F^T(d,c) = c'_F t_F^d\,, \tag{5.21}$$

$$f_E^T(d,c) = c'_E t_E^d\,, \tag{5.22}$$

$$f_N^T(d,c) = c'_N t_N^d\,. \tag{5.23}$$

Given a category space, e.g. the *Fragment Focus* category space, we define the expected number of occurrences of category i observed from the training data as:

$$E_{\tilde{p}}(f_{Fi}^{P}) = \sum_{d,c \in Observed(D,C)} \tilde{p}(d,c) f_{Fi}^{P}(d,c) = \sum_{d,c \in Observed(D,C)} \tilde{p}(d,c) c_{Fi} \,,$$

and the expected number of occurrences of category i predicted by the model as:

$$E_{p}(f_{Fi}^{P}) = \sum_{d \in D} \tilde{p}(d) \sum_{c \in C} p(c \mid d) f_{Fi}^{P}(d,c) = \sum_{d \in D} \tilde{p}(d) \sum_{c \in C} p(c \mid d) c_{Fi} \,.$$

To restrict the category distribution predicted by the model to be the same as the empirical category distribution, we define the following constraints:

$$E_{\tilde{p}}(f_{Fi}^{P}) = E_{p}(f_{Fi}^{P}), \quad 1 \le i \le 3. \tag{5.24}$$

The same types of constraints can be introduced for the *Fragment Evidence* and *Sentence Focus* category spaces:

$$E_{\tilde{p}}(f_{Ei}^{P}) = E_{p}(f_{Ei}^{P}), \; 1 \le i \le 4, \tag{5.25}$$

$$E_{\tilde{p}}(f_{Ni}^{P}) = E_{p}(f_{Ni}^{P}), \; 1 \le i \le 3. \tag{5.26}$$

Similarly, we can impose constraints over the correlation between terms and categories. Consider the *Fragment Focus* category space as an example. We define the expected number of co-occurrences between category i and term j derived from the training data as:

$$E_{\tilde{p}}(f_{Fij}^{T}) = \sum_{d,c \in Observed(D,C)} \tilde{p}(d,c) f_{Fij}^{T}(d,c) = \sum_{d,c \in Observed(D,C)} \tilde{p}(d,c) c_{Fi} t_{Fj}^{d} \,,$$

and the expected number of co-occurrences between category i and term j predicted by the model as:

$$E_{p}(f_{Fij}^{T}) = \sum_{d \in D} \tilde{p}(d) \sum_{c \in C} p(c \mid d) f_{Fij}^{T}(d,c) = \sum_{d \in D} \tilde{p}(d) \sum_{c \in C} p(c \mid d) c_{Fi} t_{Fj}^{d} \,.$$

To make the model consistent with the correlation between terms and categories as observed from the training data, we define the following constraints:

$$E_{\tilde{p}}(f_{Fij}^{T}) = E_{p}(f_{Fij}^{T}),\ 1 \le i \le 3,\ 1 \le j \le |t_F|, \tag{5.27}$$

where $|t_F|$ is the total number of terms related to the *Focus* of a fragment.

Similar constraints can be introduced for the *Fragment Evidence* and *Sentence Focus* category spaces:

$$E_{\tilde{p}}(f_{Eij}^{T}) = E_{p}(f_{Eij}^{T}), 1 \le i \le 4, 1 \le j \le |t_E|, \tag{5.28}$$

$$E_{\tilde{p}}(f_{Nij}^{T}) = E_{p}(f_{Nij}^{T}), 1 \le i \le 3, 1 \le j \le |t_N|, \tag{5.29}$$

where $|t_E|$ is the total number of terms related to the *Evidence* of a fragment; $|t_N|$ is the total number of terms related to the *Focus* of a sentence.

Considering the three distinctive characteristics of the classification, we need to impose some additional constraints on the model.

Constraints regarding the Multi-label Classification

To solve the multi-label classification problem, we use the approach introduced by Zhu *et al.* [ZJXG05]: given a category vector, $c \equiv (c_1, ..., c_n)$, the Maximum Entropy model is restricted to comply with the second-order statistical property $c_i c_j$ of the training examples. More specifically, if the co-occurrence frequency of category i and j is high, i.e. the product of the ith and jth elements of the category vector *c* is 1 for most of the examples, the expected value of $c_i c_j$ over the training data should be relatively high.

Consequently, the expected value of $c_i c_j$ predicted by the model should be high. The constraints can be expressed as the following equation:

$$E_{\tilde{p}}(c_i c_j) = E_p(c_i c_j)\,.$$

In our model, consider the *Fragment Focus* category space as an example. We can introduce a feature function matrix f_F^M to represent the co-occurrence among categories within the category space. For each fragment-label pair (d, c), the feature function matrix f_F^M is defined as the product of the category vector c_F and its transpose c'_F:

$$f_F^M(d, c) = c'_F c_F\,. \tag{5.30}$$

f_F^M is a 3×3 symmetric matrix, where the Boolean element f_{Fij}^M indicates whether categories *i* and *j* co-occur.

Based on the feature function matrix f_F^M, the correlation among categories derived from training examples can be measured by a 3×3 symmetric matrix, as illustrated in Figure 5.3.1. The element located in the ith row and the jth column measures the expected number of co-occurrences of categories *i* and j, defined by:

$$E_{\tilde{p}}(f_{Fij}^M) = \sum_{d,c \in Observed(D,C)} \tilde{p}(d,c)\, f_{Fij}^M(d,c) = \sum_{d,c \in Observed(D,C)} \tilde{p}(d,c) c_{Fi} c_{Fj}\,.$$

Suppose that a considerable number of fragments belong to both of the categories *Scientific* and *Methodology*, then the value of the element located in the 1st row and the 3rd column (or the 3rd row and the 1st column) in the correlation matrix should be relatively high.

	S	G	M
S	x_1	x_4	x_6
G	x_4	x_2	x_5
M	x_6	x_5	x_3

High value indicates high correlation between the categories *Scientific* and *Methodology*

Figure 5.3.1. The correlation matrix for categories within the Fragment Focus category space.

We restrict each element value in the correlation matrix predicted by the model,

$E_p(f_{Fij}^M) = \sum_{d \in D} \tilde{p}(d) \sum_{c \in C} p(c \mid d) f_{Fij}^M(d,c) = \sum_{d \in D} \tilde{p}(d) \sum_{c \in C} p(c \mid d) c_{Fi} c_{Fj}$, to be consistent with that derived from the training examples by adding the constraint[4]:

$$E_{\tilde{p}}(f_{Fij}^M) = E_p(f_{Fij}^M)\,,\ 1 \le i < j \le 3\,. \tag{5.31}$$

Similarly, for a given fragment-label pair (d,c), two other feature function matrices can be introduced to capture the correlation among categories within the *Fragment Evidence* and *Sentence Focus* category spaces respectively:

$$f_E^M(d,c) = c_E' c_E\,, \tag{5.32}$$

$$f_N^M(d,c) = c_N' c_N\,. \tag{5.33}$$

Accordingly, two other constraints are imposed on the model:

$$E_{\tilde{p}}(f_{Eij}^M) = E_p(f_{Eij}^M)\,,\ 1 \le i < j \le 4\,, \tag{5.34}$$

$$E_{\tilde{p}}(f_{Nij}^M) = E_p(f_{Nij}^M)\,,\ 1 \le i < j \le 3\,. \tag{5.35}$$

Constraints regarding the Correlation between Focus and Evidence

To capture the co-occurrence between categories from the *Fragment Focus* and *Fragment Evidence* category spaces, we can introduce a feature function matrix, f^R. For each

[4] Since the matrix is symmetric, we only need to impose constraints on the elements above the diagonal.

fragment-label pair (d, c), the feature function matrix f^R is defined as the product of the category vectors c'_F and c_E:

$$f^R(d,c) = c'_F c_E \,. \tag{5.36}$$

f^R is a 3×4 matrix, where the Boolean element f^R_{ij} indicates whether category *i* from the *Focus* dimension and category *j* from the *Evidence* dimension co-occur.

With the feature function matrix f^R, for a given set of training examples, the correlation between the *Focus* and *Evidence* dimensions can be viewed as a 3×4 matrix as shown in Figure 5.3.2. Each row represents a category from the *Focus* dimension. Each column represents a category from the *Evidence* dimension. The element located in the ith row and the jth column measures the expected number of co-occurrences of category *i* from the *Focus* dimension and category *j* from the *evidence* dimension, defined by:

$$E_{\tilde{p}}(f^R_{ij}) = \sum_{d,c \in Observed(D,C)} \tilde{p}(d,c) f^R_{ij}(d,c) = \sum_{d,c \in Observed(D,C)} \tilde{p}(d,c) c_{Fi} c_{Ej} \;.$$

For example, if there exists high correlation between the categories *Methodology* and *Explicit evidence*, namely, the *Evidence* level is *Explicit evidence (E3)* for most of the fragments whose *Focus* is *Methodology*, then the value of the element located in the 3rd row and the 4th column of the correlation matrix should be relatively high.

	E0	E1	E2	E3
S	x	x	x	x
G	x	x	x	x
M	x	x	x	x

High value indicates strong correlation between the categories *Methodology* and *Explicit evidence*

Figure 5.3.2. The correlation matrix for the Fragment Focus and Fragment Evidence category spaces.

To restrict the model to produce the same value for each element in the correlation matrix as derived from the training examples, a constraint is added over the two category spaces:

$$E_{\tilde{p}}(f_{ij}^{R}) = E_{p}(f_{ij}^{R}),\ 1 \le i \le 3,\ 1 \le j \le 4, \tag{5.37}$$

where $E_{p}(f_{ij}^{R}) = \sum_{d \in D} \tilde{p}(d) \sum_{c \in C} p(c \mid d) f_{ij}^{R}(d,c) = \sum_{d \in D} \tilde{p}(d) \sum_{c \in C} p(c \mid d) c_{Fi} c_{Ej}$.

Constraints regarding the Context Dependent Classification

To capture the co-occurrence between categories from the *Fragment Focus* and *Sentence Focus* category spaces, we can introduce a feature function matrix f^{N}. For each fragment-label pair (d,c), the feature function matrix f^{N} is defined as the product of the category vectors c'_{F} and c_{N}:

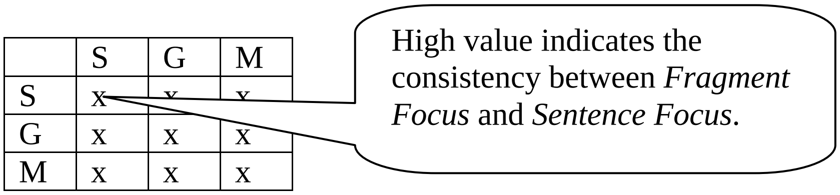
$$f^{N}(d,c) = c'_{F} c_{N}. \tag{5.38}$$

f^{N} is a 3×3 matrix, where the Boolean element f_{ij}^{N} indicates whether there exists co-occurrence between *Fragment Focus* category *i* and *Sentence Focus* category *j*.

With the feature function matrix f^{N}, for a given set of training examples, the correlation between the two category spaces, *Fragment Focus* and *Sentence Focus*, can be measured by a 3×3 matrix as shown in Figure 5.3.3. The rows correspond to the *Fragment Focus* category space. The columns correspond to the *Sentence Focus* category space. The element located in the i^th^ row and the j^th^ column measures the expected number of co-occurrences of *Fragment Focus* category *i* and *Sentence Focus* category *j*, defined by:

$$E_{\tilde{p}}(f_{ij}^{N}) = \sum_{d,c \in Observed(D,C)} \tilde{p}(d,c) f_{ij}^{N}(d,c) = \sum_{d,c \in Observed(D,C)} \tilde{p}(d,c) c_{Fi} c_{Nj} \ .$$

The element value gives a quantitative measurement of how high the correlation is between the categories. By the definition of the *Sentence Focus* at the beginning of Section 5.3.1, the diagonal element value in the correlation matrix should be relatively high. That is, the *Focus* of a fragment tends to be consistent with the *Focus* of the sentence where it comes from. If in the training dataset many fragments with different *Focus*, say *Scientific* or *Methodology*, co-occur in the same sentences, in the correlation matrix, the element located in the 1st row and the 3rd column, and the element located in the 3rd row and the 1st column, can also have high values.



	S	G	M
S	x	x	x
G	x	x	x
M	x	x	x

High value indicates the consistency between *Fragment Focus* and *Sentence Focus*.

Figure 5.3.3. The correlation matrix for the Fragment Focus and Sentence Focus category spaces.

To make use of the dependence of *Fragment Focus* on *Sentence Focus*, we need to restrict the model to produce the same value for each element in the correlation matrix as derived from the training examples. We therefore add a constraint over the category spaces *Fragment Focus* and *Sentence Focus*:

$$E_{\tilde{p}}(f_{ij}^{N}) = E_{p}(f_{ij}^{N}),\ 1 \le i \le 3,\ 1 \le j \le 3, \qquad (5.39)$$

where $E_{p}(f_{ij}^{N}) = \sum_{d \in D} \tilde{p}(d) \sum_{c \in C} p(c \mid d) f_{ij}^{N}(d,c) = \sum_{d \in D} \tilde{p}(d) \sum_{c \in C} p(c \mid d) c_{Fi} c_{Nj} \, .$

5.3.4 The Parametric Form of the Model

In the previous section, we have defined five types of feature matrices: the occurrence of categories within each category space, f_F^P, f_E^P, f_N^P; the correlation between terms and categories within each category space, f_F^T, f_E^T, f_N^T; the correlation between categories within each category space, f_F^M, f_E^M, f_N^M; the correlation between the *Fragment Focus* and *Fragment Evidence* category spaces, f^R; and the correlation between the *Fragment Focus* and *Sentence Focus* category spaces f^N.

Based on the above features, we need to introduce five types of model parameters that reflect the contributions of the features toward the final classification decision: parameters $\lambda_F^P, \lambda_E^P, \lambda_N^P$ represent the weights of the feature matrices f_F^P, f_E^P, f_N^P respectively; parameters $\lambda_F^T, \lambda_E^T, \lambda_N^T$ represent the weights of the feature matrices f_F^T, f_E^T, f_N^T respectively; parameters $\lambda_F^M, \lambda_E^M, \lambda_N^M$ represent the weights of the feature matrices f_F^M, f_E^M, f_N^M respectively; parameter λ^R represents the weight of the feature matrix f^R; parameter λ^N represents the weight of the feature matrix f^N. Thus, the optimal conditional probability $p^*(c \mid d)$ has the following parametric form (Refer to Appendix D for the detailed derivations):

$$p^*(c \mid d) = \frac{1}{Z(d)} \exp(\lambda_F^P \cdot f_F^P + \lambda_E^P \cdot f_E^P + \lambda_N^P \cdot f_N^P + \lambda_F^M \cdot f_F^M + \lambda_E^M \cdot f_E^M + \lambda_N^M \cdot f_N^M$$

$$+ \lambda^R \cdot f^R + \lambda^N \cdot f^N + \lambda_F^T \cdot f_F^T + \lambda_E^T \cdot f_E^T + \lambda_N^T \cdot f_N^T), \tag{5.40}$$

where $\lambda \cdot f$ denotes the sum of the pairwise products of the elements in the two matrices, λ and f, formally: $\lambda \cdot f = \sum_{i,j} \lambda_{ij} f_{ij}$, and Z(d) is the normalization factor, defined as :

$$Z(d) = \sum_{c \in C} \exp(\lambda_F^P \cdot f_F^P + \lambda_E^P \cdot f_E^P + \lambda_N^P \cdot f_N^P + \lambda_F^M \cdot f_F^M + \lambda_E^M \cdot f_E^M + \lambda_N^M \cdot f_N^M$$

$$+ \lambda^R \cdot f^R + \lambda^N \cdot f^N + \lambda_F^T \cdot f_F^T + \lambda_E^T \cdot f_E^T + \lambda_N^T \cdot f_N^T). \qquad (5.41)$$

We use the IIS algorithm to find the optimal parameter sets. We start from an arbitrary initial parameter set $\Lambda \equiv (\lambda_F^P, \lambda_E^P, \lambda_N^P, \lambda_F^M, \lambda_E^M, \lambda_N^M, \lambda^R, \lambda^N, \lambda_F^T, \lambda_E^T, \lambda_N^T)$, and at each step, find an improvement, $\delta \equiv (\delta\lambda_F^P, \delta\lambda_E^P, \delta\lambda_N^P, \delta\lambda_F^M, \delta\lambda_E^M, \delta\lambda_N^M, \delta\lambda^R, \delta\lambda^N, \delta\lambda_F^T, \delta\lambda_E^T, \delta\lambda_N^T)$, such that the new model $\Lambda + \delta$ yields a higher log likelihood with respect to the training data (Refer to Appendix D for the detailed formulations).

To classify a data instance, we enumerate all the possible combinations of the three category vectors c_F, c_E, c_N, and calculate their conditional probabilities. The data instance would be assigned to the class label that yields the highest probability.

So far, we have designed a model based on Maximum Entropy to address the special needs of the classification along the *Focus* and *Evidence* dimensions. The classification performance of this model along the two dimensions, as well as the experiments with Naïve Bayes and SVM classifiers along other dimensions, are presented in the next chapter.

Chapter 6

Classification Performance: Evaluation and Analysis

In this chapter, we investigate the feasibility of using machine learning methods to automatically perform the fragment annotation task. We examine three classification algorithms: Naïve Bayes, Support Vector Machines (SVMs), and Maximum Entropy[1]. The classification performance is evaluated separately for each of the five dimensions: *Focus, Evidence, Certainty, Polarity* and *Trend*. We present the experimental results, and discuss issues and solutions.

6.1 Dataset

Our dataset consists of sentences sampled from different sections of full-text journal articles and from Medline abstracts. The dataset has three parts, and each part was annotated by a different group of three annotators. Part 1 contains 200 sentences annotated by the three authors of the Annotation Guidelines [ShWR06]. Parts 2 and 3 contain 625 sentences each, annotated by annotators (advanced graduate students in Biology) who were trained using the guidelines. We experiment with two approaches to decide on the fragmentation and annotation of a sentence.

To ensure the quality of the training data, for each dimension, we first use sentences whose annotated labels along that dimension are agreed upon by at least two annotators for part 1, and sentences whose annotated labels are agreed upon by all of the

[1] In early stages of this work, we also conducted experiments using other classification methods, such as Decision Trees and Neural Networks, which produced relatively inferior results.

three annotators for parts 2 and 3. Since we have designed a Maximum Entropy model to classify data along the *Focus* and *Evidence* dimensions simultaneously, for these two dimensions, we generate a set of sentences with the annotated labels agreeing on both of the two dimensions. As a result, four datasets were generated for the five dimensions. Table 6.1.1 shows the number of sentences and the number of fragments of each dataset.

However, as can be seen from Table 6.1.1, the training data for each dimension is still limited. As further investigation, we generate more training data by selecting sentences based on the majority agreement for all of the manually annotated examples. That is, for each dimension, we use sentences whose annotated labels along that dimension are agreed upon by at least two annotators. In addition to the three parts of the dataset mentioned above, we add a new dataset[2] which contains 625 sentences. As a result, we generate four larger datasets whose properties are shown in Table 6.1.2.

Table 6.1.1. The dataset generated for each classification dimension.

	Focus and Evidence	Certainty	Polarity	Trend
Number of Sentences	296	796	916	822
Number of Fragments	354	876	1031	919
Dataset Name	*Frag_FE*	*Frag_C*	*Frag_P*	*Frag_T*

Table 6.1.2. The datasets generated based on the majority annotation agreement.

	Focus and Evidence	Certainty	Polarity	Trend
Number of Sentences	1051	1671	1819	1739
Number of Fragments	1168	1902	2100	2003
Dataset Name	*Frag_M_FE*	*Frag_M_C*	*Frag_M_P*	*Frag_M_T*

[2] This dataset is annotated by only two annotators, thereby it cannot be used when we select sentences whose labels are agreed upon by three annotators.

The main classification experiments in this thesis are performed on the dataset shown in Table 6.1.1. Experimental results and analysis are provided in Section 6.3. The classification performance on the larger datasets together with basic analysis are reported in Section 6.4. Before examining the experimental results, we first discuss the measures that are used to evaluate the classification performance.

6.2 Performance Measures for the Fragment Classification

To evaluate the classification performance, we use the standard measures in text categorization tasks, *Precision* and *Recall*, as discussed in Section 2.2.3. Specifically, for a given category *i*, *Precision* and *Recall* are defined as:

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \quad , \; Recall_i = \frac{TP_i}{TP_i + FN_i} .$$

The confusion matrix of a classifier provides a more straightforward way to understand the definitions of *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), and *False Negative* (FN) with respect to a category. For example, suppose we have three categories, denoted as *A*, *B*, *C* respectively. Table 6.2.1 shows the entries in the confusion matrix that contribute to TP_A, FP_A, TN_A and FN_A respectively. Each row represents a true category, and each column represents a predicted category.

Table 6.2.1. The entries that contribute to TP_A*,* FP_A*,* TN_A and FN_A *in the confusion matrix for a three-category case.*

	A	B	C
A	TP_A	FN_A	FN_A
B	FP_A	TN_A	TN_A
C	FP_A	TN_A	TN_A

The standard *Precision* and *Recall* are initially defined for single-label classification tasks. The *True Positive*, *False Positive*, *True Negative*, and *False Negative* examine whether a data example is correctly classified into – or rejected from – a category, without considering the issue of partially correct category assignment when a data example is associated with multiple categories. In the case when the classification is multi-labeled, all the combined multiple categories are treated as independent new categories, namely, the multi-label classification is converted into a single-label one.

In our fragment annotation task, the classification along the *Focus* and *Evidence* dimensions is multi-labeled. Consider the *Focus* dimension as an example. There are three basic categories: *Scientific*, denoted as *S*; *Generic*, denoted as *G*; and *Methodology*, denoted as *M*. These lead to seven possible combinations: *S*, *G*, *M*, *MS*, *MG*, *SG*, and *MSG*. The standard *Precision* and *Recall* can be obtained by treating all of the seven combinations as independent categories (Table 6.2.2). As can be seen from the confusion matrix shown in Table 6.2.2, both *Precision* and *Recall* for the category *MS* are zero. Although the examples under this category are either classified as *M* or *S*, they are treated as completely misclassified examples. Because the standard *Precision* and *Recall* do not take into account the partial success in category assignment, they cannot accurately evaluate the performance of our classification task.

Table 6.2.2. An example confusion matrix that ignores the partially correct category assignment.

	S	G	M	MS	MG	SG	MSG	Precision	Recall
S	267	4	1	0	0	0	0	0.86	0.98
G	26	9	1	0	0	0	0	0.69	0.25
M	13	0	24	0	0	0	0	0.80	0.65
MS	3	0	4	0	0	0	0	0.00	0.00
MG	0	0	0	0	0	0	0	NA	NA
SG	2	0	0	0	0	0	0	0.00	0.00
MSG	0	0	0	0	0	0	0	NA	NA
Average								0.47	0.38

Based on the nature of our categories, we propose an alternative performance measure, modifying the standard *Precision* and *Recall*, such that partial success in a multiple category assignment can be rewarded. We next go over this measure in detail. Since *Precision* and *Recall* are only concerned with *True Positive*, *False Positive*, and *False Negative* with respect to a category, we only consider TP_i, FP_i, and FN_i with respect to category *i* in the following discussion. Our performance measure for the multi-labeled classification follows the standard definitions of *Precision* and *Recall*, but uses different ways to calculate *True Positive*, *False Positive*, and *False Negative* with respect to a category.

As previously discussed in Section 5.3.2, we can use a Boolean vector $c \equiv (c_1,...,c_m)$ to represent the label of a fragment, where m is the total number of categories. Each element c_i indicates the membership of the fragment in category *i*, defined as:

$$c_i = \begin{cases} 1 \;\; \textit{if the fragment is assigned to category } i; \\ 0 \;\; \textit{otherwise.} \end{cases}$$

We use $c^{dT} \equiv (c_{1}^{dT},...,c_{m}^{dT})$ to denote the true label of a fragment *d*, and $c^{dP} \equiv (c_{1}^{dP},...,c_{m}^{dP})$ to denote its predicted label. Suppose we view the total value of the classification of a fragment as 1. If a fragment is assigned into *k* categories, each category thus gets $\frac{1}{k}$ of this total value. For the classification of each fragment, we refer to the value that category *i* should get as the *True Contribution* of the fragment to category *i*, and the value that is assigned to category *i* by a classifier as the *Predicted Contribution* of the fragment to category *i*. We can calculate the *True Contribution* of a fragment to each category by normalizing the category vector c^{dT} :

$$TC^{d} \equiv (TC_1^d, ..., TC_m^d) = (\frac{c_{1}^{dT}}{\sum_{i=1}^{m} c_i^{dT}}, ..., \frac{c_{m}^{dT}}{\sum_{i=1}^{m} c_i^{dT}}).$$

Similarly, the *Predicted Contribution* of a fragment to each category can be calculated by normalizing c^{dP}:

$$PC^{d} \equiv (PC_1^d, ..., PC_m^d) = (\frac{c_{1}^{dP}}{\sum_{i=1}^{m} c_i^{dP}}, ..., \frac{c_{m}^{dP}}{\sum_{i=1}^{m} c_i^{dP}}).$$

We use δ_i^{dTP} , δ_i^{dFP} , and δ_i^{dFN} to represent the changes in *True Positive*, *False Positive*, and *False Negative* with respect to category *i* when introducing fragment *d* into the dataset.

δ_i^{dTP} is calculated based on the agreement between the *True Contribution* and the *Predicted Contribution* of a fragment to category *i*, namely, the value that is correctly assigned to category *i*, defined as*:*

$$\delta_i^{dTP} = \min(TC_i^d, PC_i^d).$$

δ_i^{dFP} and δ_i^{dFN} are calculated based on the difference between the *True Contribution* and the *Predicted Contribution* of a fragment to category *i*. If the *Predicted Contribution* of a fragment to category *i* is larger than its *True Contribution*, namely, the value that is assigned to category *i* is incorrectly higher, the difference between the *Predicted Contribution* and the *True Contribution* is considered as *False Positive*, defined as:

$$\delta_i^{dFP} = \begin{cases} PC_i^d - TC_i^d, \ if \ PC_i^d > TC_i^d; \\ 0 \ otherwise. \end{cases}$$

If the *True Contribution* of a fragment to category *i* is larger than its *Predicted Contribution*, namely, the value that is assigned to category *i* is incorrectly lower, the difference between the *True Contribution* and the *Predicted Contribution* is considered as *False Negative*, defined as:

$$\delta_i^{dFN} = \begin{cases} TC_i^d - PC_i^d, \ if \ TC_i^d > PC_i^d; \\ 0 \ otherwise. \end{cases}$$

Thus, the *True Positive*, *False Positive*, and *False Negative* with respect to category *i*, i.e., TP_i, FP_i, and FN_i, can be calculated by summing over the changes δ_i^{dTP}, δ_i^{dFP}, and δ_i^{dFN} brought by each fragment in the training set.

To illustrate how our performance measure is defined, we consider the *Focus* dimension as an example. As previously mentioned, there are three basic categories: *Scientific*, denoted as *S*; *Generic*, denoted as *G*; and *Methodology*, denoted as *M*. We use $c \equiv (c_S, c_G, c_M)$ to denote the *Focus* label of a fragment. Specifically, if the true category of a fragment *d* is *S*, and the predicted category is *MS*, the *True Contribution* of the fragment to the three categories can be represented as a vector, $TC^d = (1,0,0)$, and the *Predicted Contribution* can be represented as a vector, $PC^d = (0.5,0,0.5)$. Accordingly, the changes in TP_S , FP_S , and FN_S with respect to category *S* by introducing the fragment into the dataset are:

$$\delta_S^{dTP} = \min(1,0.5) = 0.5\,,$$

$$\delta_S^{dFP} = 0\,,$$

$$\delta_S^{dFN} = 1 - 0.5 = 0.5\,.$$

The changes in TP_M , FP_M , and FN_M with respect to category *M* are:

$$\delta_M^{dTP} = \min(0,0.5) = 0\,,$$

$$\delta_M^{dFP} = 0.5 - 0 = 0.5\,,$$

$$\delta_M^{dFN} = 0\,.$$

That is, the *True Positive* and *False Negative* with respect to category *S*, namely, TP_S and FN_S are increased by 0.5 respectively; and the *False Positive* with respect to category *M* is increased by 0.5. Thus, the partially correct assignment of fragment *d* under category *S* is rewarded (TP_S is increased by 0.5). We illustrate the detailed calculation of TP , FP , and FN with respect to each category in Table 6.2.3. Each row represents the true categories of an example, each column represents the predicted categories, and each element represents a possible way to classify an example.

Table 6.2.3. The calculation of TP , FP , and FN with respect to each category for any possible classification of an example. S denotes ***Scientific****, G denotes* ***Generic****, M denotes* ***Methodology****, MS, MG, SG, and MSG denote the combinations of them.*

	S	G	M	MS	MG	SG	MSG
S	$TP_S + 1$	$FN_S + 1$ $FP_G + 1$	$FN_S + 1$ $FP_M + 1$	$TP_S + 0.5$ $FN_S + 0.5$ $FP_M + 0.5$	$FN_S + 1$ $FP_M + 0.5$ $FP_G + 0.5$	$TP_S + 0.5$ $FN_S + 0.5$ $FP_G + 0.5$	$TP_S + 0.33$ $FN_S + 0.67$ $FP_M + 0.33$ $FP_G + 0.33$
G	$FP_S + 1$ $FN_G + 1$	$TP_G+ 1$	$FP_M + 1$ $FN_G + 1$	$FP_S + 0.5$ $FP_M + 0.5$ $FN_G + 1$	$FP_M + 0.5$ $TP_G + 0.5$ $FN_G + 0.5$	$FP_S + 0.5$ $TP_G + 0.5$ $FN_G + 0.5$	$FP_S + 0.33$ $FP_M + 0.33$ $TP_G + 0.33$ $FN_G + 0.67$
M	$FP_S + 1$ $FN_M + 1$	$FN_M + 1$ $FP_G + 1$	$TP_M + 1$	$FP_S + 0.5$ $TP_M + 0.5$ $FN_M + 0.5$	$TP_M + 0.5$ $FN_M + 0.5$ $FP_G + 0.5$	$FP_S + 0.5$ $FN_M + 1$ $FP_G + 0.5$	$FP_S + 0.33$ $TP_M + 0.33$ $FN_M + 0.67$ $FP_G + 0.33$
MS	$TP_S + 0.5$ $FP_S + 0.5$ $FN_M + 0.5$	$FN_S + 0.5$ $FN_M + 0.5$ $FP_G + 1$	$FN_S + 0.5$ $TP_M + 0.5$ $FP_M + 0.5$	$TP_S + 0.5$ $TP_M + 0.5$	$FN_S + 0.5$ $TP_M + 0.5$ $FP_G + 0.5$	$TP_S + 0.5$ $FN_M + 0.5$ $FP_G + 0.5$	$TP_S + 0.33$ $FN_S + 0.17$ $TP_M + 0.33$ $FN_M + 0.17$ $FP_G + 0.33$
MG	$FP_S + 1$ $FN_M + 0.5$ $FN_G + 0.5$	$FN_M + 0.5$ $TP_G + 0.5$ $FP_G + 0.5$	$TP_M + 0.5$ $FP_M + 0.5$ $FN_G + 0.5$	$FP_S + 0.5$ $TP_M + 0.5$ $FN_G + 0.5$	$TP_M + 0.5$ $TP_G + 0.5$	$FP_S + 0.5$ $FN_M + 0.5$ $TP_G + 0.5$	$FP_S + 0.33$ $TP_M + 0.33$ $FN_M + 0.17$ $TP_G + 0.33$ $FN_G + 0.17$
SG	$TP_S + 0.5$ $FP_S + 0.5$ $FN_G + 0.5$	$FN_S + 0.5$ $TP_G + 0.5$ $FP_G + 0.5$	$FN_S + 0.5$ $FP_M + 1$ $FN_G + 0.5$	$TP_S + 0.5$ $FP_M + 0.5$ $FN_G + 0.5$	$FN_S + 0.5$ $FP_M + 0.5$ $TP_G + 0.5$	$TP_S + 0.5$ $TP_G + 0.5$	$TP_S + 0.33$ $FN_S + 0.17$ $FP_M + 0.33$ $TP_G + 0.33$ $FN_G + 0.17$
MSG	$TP_S + 0.33$ $FP_S + 0.67$ $FN_M + 0.33$ $FN_G + 0.33$	$FN_S + 0.33$ $FN_M + 0.33$ $TP_G + 0.33$ $FP_G + 0.67$	$FN_S + 0.33$ $TP_M + 0.33$ $FP_M + 0.67$ $FN_G + 0.33$	$TP_S + 0.33$ $FP_S + 0.17$ $TP_M + 0.33$ $FP_M + 0.17$ $FN_G + 0.33$	$FN_S + 0.33$ $TP_M + 0.33$ $FP_M + 0.17$ $TP_G + 0.33$ $FP_G + 0.17$	$TP_S + 0.33$ $FP_S + 0.17$ $FN_M + 0.33$ $TP_G + 0.33$ $FP_G + 0.17$	$TP_S + 0.33$ $TP_M + 0.33$ $TP_G + 0.33$

Based on the new definitions of *True Positive*, *False Positive*, and *False Negative*, the recalculated confusion matrix, as well as *Precision* and *Recall* that take into account the partial success in category assignment for multi-label classification, are shown in Table 6.2.4. Compared to the original confusion matrix that treats all the combined multiple categories as independent new categories (Table 6.2.2), the average *Precision* and *Recall* increase from 0.47 and 0.38, to 0.81 and 0.62, respectively. Based on this new measure, the performance of our multi-label classification task can be more accurately evaluated.

Table 6.2.4. The recalculated confusion matrix where the partial successful category assignment for multi-label classification is rewarded.

	S	G	M	Precision	Recall
S	269.5	4.0	3.0	0. 87	0.98
G	27.0	9.0	1.0	0. 69	0.24
M	14.5	0.0	26.0	0. 87	0.64
Average				0. 81	0.62

6.3 Experimental Results and Analysis

We evaluate the classification performance along each dimension separately. Since most of the work in this thesis is concerned with the design of a classification model for the *Focus* and *Evidence* dimensions, which involves several distinctive characteristics compared to general text classification tasks, we mainly focus on the experimental results and analysis on these two dimensions. We then report the classification performance on the other three dimensions and present preliminary analysis.

6.3.1 Focus and Evidence

Since our Maximum Entropy model classifies data along the *Focus* and *Evidence* dimensions simultaneously, we examine the classification performance on the dataset *Frag_FE*, which consists of fragments whose annotated labels agree on both of these dimensions.

The Distribution of Fragments among Categories

Before examining the classification performance, we first look at the distribution of fragments in the categories, which is shown in Figures 6.3.1 and 6.3.2. We can see that the distribution along the *Focus* dimension is skewed. Most of the fragments belong to the *Scientific* category, while only a few belong to multiple categories, i.e., *Methodology* and *Scientific*, or *Scientific* and *Generic*. In contrast, the distribution of fragments among categories along the *Evidence* dimension is more uniform. However, no examples with multiple *Evidence* levels are included. Since the annotation for the examples consists of the majority or the unanimous annotation agreement, such agreement is less likely to occur on multiply annotated categories than on a single category assignment. Therefore, only a few examples with multiple category assignment exist in our dataset.

A similar problem exists in the sentence fragmentation. As can be seen from Table 6.1.1, the total number of sentences and the total number of fragments are very close to each other within each dataset, which means that the fraction of sentences with multiple fragments is small. The reason is that most of the long sentences with several fragments, heterogeneous in content, are filtered out due to the disagreement between annotators. Further revision of the manual annotation is required to improve the quality of the dataset such that it can more accurately reflect the real characteristics of scientific literature.

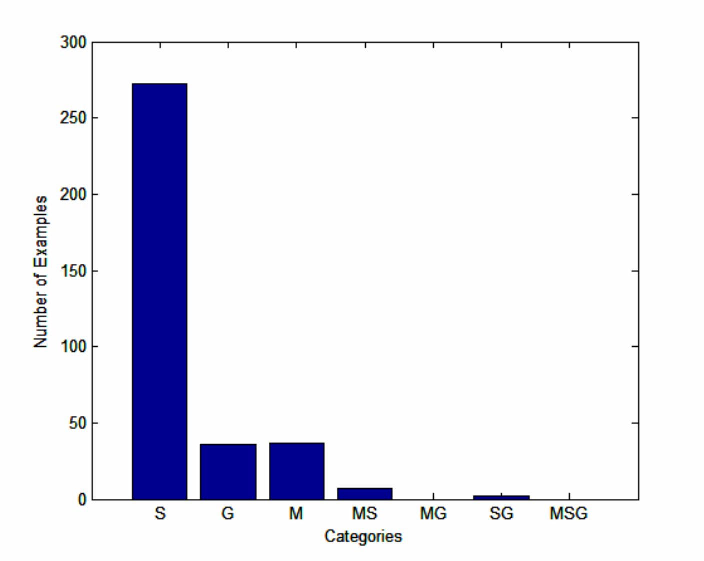


Figure 6.3.1. The distribution of fragments in the categories along the Focus dimension over the dataset Frag_FE. S denotes ***Scientific****, G denotes* ***Generic****, M denotes* ***Methodology****, MS, MG, SG, and MSG denote their combinations.*

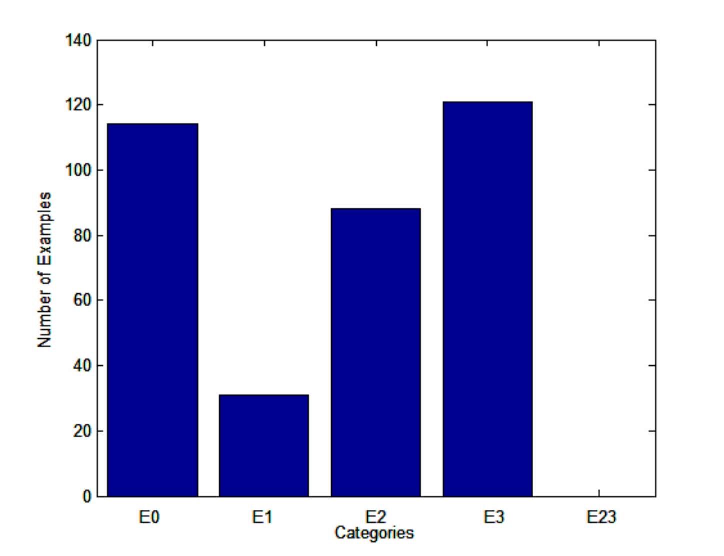


Figure 6.3.2. The distribution of fragments in the categories along the Evidence dimension over the dataset Frag_FE. E0 denotes ***No evidence****, E1 denotes* ***Claim of evidence without verifying information****, E2 denotes* ***Explicit citation****, E3 denotes* ***Explicit evidence****, and E23 denotes both* ***Explicit citation*** *and* ***Explicit evidence****.*

Experimental Results

Before applying the classification algorithms, we need to preprocess the data. We follow the preprocessing procedures as discussed in Section 4.1.

First, each sentence fragment is converted into a vector of term weights, where terms are formed by individual words and *n*-grams. As discussed in Section 4.1, there are two ways to generate *n*-grams: treating *stop words* as *n*-gram boundaries, which produces shorter word sequences; and ignoring *stop words* in *n*-gram generation, which produces longer word sequences. In our experiments, the two approaches yield similar classification performance. Here we only report the experimental results based on the latter, i.e., ignoring *stop words* in *n*-gram generation.

Next, the dimension reduction procedures introduced in Section 4.2 are applied. After removing *stop words*, filtering out terms occurring less than two times, and selecting words based on their POS tags, about 800 terms are retained in the dataset *Frag_FE* (354 fragments). To further investigate the effect of the number of terms on the classification performance, we experiment with the term selection functions surveyed in Section 2.2.1. In our experiment, the terms for the *Focus* dimension are first ranked according to their scores based on the term selection function *chi-square*. Then the performance of a Naïve Baiyes classifier is investigated on thirteen types of fragment representation where the number of terms varies from 20 to 810. The result is shown in Figure 6.3.3. We can see that the highest prediction accuracy is obtained when the number of terms is between 20 and 50. Although there are variations in performance with the addition of terms (e.g., slightly improved accuracy is observed when the number of terms increases from 500 to 700), in general, the inclusion of more terms degrades the performance.

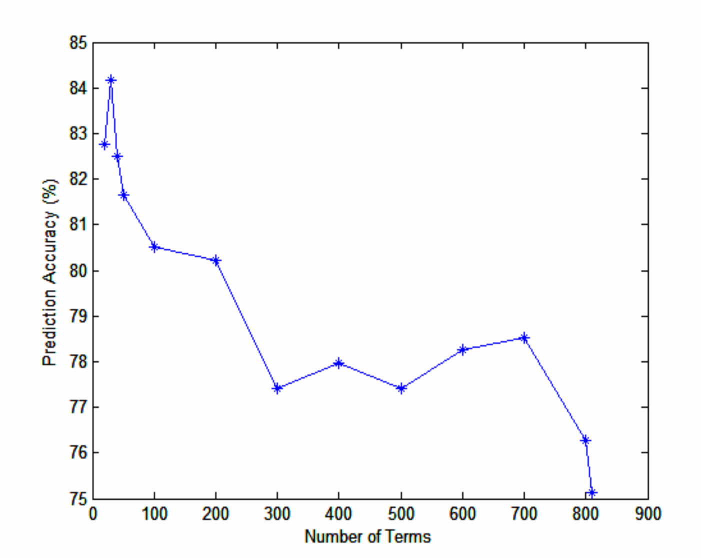


Figure 6.3.3. The performance of a Naïve Bayes classifier as a function of the number of terms used.

A similar experiment is performed on the *Evidence* dimension, and the best performance is obtained when the number of terms is between 100 and 200. Therefore, the top 30 terms are selected to generate the fragment representation for the *Focus* dimension, and the top 200 terms are used for the *Evidence* dimension.

Three classifiers, Naïve Bayes [WiFr05], Support Vector Machine (SVM) [ChLi01], and Maximum Entropy are investigated. Classification performance is measured in terms of *Precision*, *Recall* and *F-measure* (partially correct category assignment is rewarded). We also provide the classification accuracy[3] as a reference.

[3] The accuracy takes into account the partial successful category assignment for multi-label classification. It can be calculated as follows: Accuracy = $\frac{\sum_{i}^{m} TP_i}{|D|}$, where m is the total number of categories, and $|D|$ is the total number of examples. The calculation of TP_i with respect to category *i* is introduced in Section 6.2.

Tables 6.3.1 and 6.3.2 show the performance of each classifier in the 5-fold cross validation (CV) test mode[4]. To ensure that the classification performance under the 5-fold CV mode is relatively stable, we test five different random splits. Based on each split, we measure the performance of the Maximum Entropy model under the 5-fold CV test mode. The standard deviation for the *F-measures* from the five different splits is 0.01 along the *Focus* dimension, and 0.007 along the *Evidence* dimension. These results indicate that the classification performance does not vary much with different splits. We report the preliminary results based on our small dataset as a reference to investigate the feasibility of automatic fragment annotation, rather than to provide performance comparison between different classification models. A much larger dataset is required to conduct reliable performance comparisons in the future.

Table 6.3.1. The performance of the Focus classification on the dataset Frag_FE. The dataset contains 296 sentences (354 fragments). S denotes ***Scientific****, G denotes* ***Generic****, and M denotes* ***Methodology****.*

	Precision			Recall			F-measure			Accuracy		
Category	Naïve Bayes	SVM	Maximum Entropy	Naïve Bayes	SVM	Maximum Entropy	Naïve Bayes	SVM	Maximum Entropy	Naïve Bayes	SVM	Maximum Entropy
S	0.86	0.84	0.87	0.99	0.99	0.98	0.92	0.91	0.92	0.86	0.84	0.86
G	0.73	0.80	0.69	0.22	0.11	0.24	0.33	0.19	0.36			
M	0.88	0.86	0.87	0.52	0.52	0.64	0.65	0.65	0.74			
Average	0.82	**0.83**	0.81	0.58	0.54	**0.62**	0.68	0.65	**0.70**			

[4] All the experiments in this chapter are conducted in a 5-fold cross validation (CV) test mode.

Table 6.3.2. The performance of the Evidence classification on the dataset Frag_FE. The dataset contains 296 sentences (354 fragments). E0 denotes **No evidence***, E1 denotes* **Claim of evidence without verifying information***, E2 denotes* **Explicit citation***, and E3 denotes* **Explicit evidence***.*

Category	Precision			Recall			F-measure			Accuracy		
	Naïve Bayes	SVM	Maximum Entropy	Naïve Bayes	SVM	Maximum Entropy	Naïve Bayes	SVM	Maximum Entropy	Naïve Bayes	SVM	Maximum Entropy
E0	0.79	0.75	0.76	0.98	0.92	0.96	0.87	0.82	0.85	0.86	0.83	0.85
E1	0.93	0.90	0.85	0.52	0.63	0.41	0.67	0.74	0.55			
E2	0.92	0.94	0.94	0.91	0.88	0.88	0.92	0.91	0.91			
E3	0.89	0.81	0.89	0.77	0.71	0.80	0.83	0.76	0.84			
Average	**0.88**	0.85	0.86	**0.80**	0.78	0.76	**0.84**	0.82	0.81			

We also investigated other term space reduction functions such as *Information gain* and *Odds ratio* [YaPe97, Seba99], as well as the term exaction (term synthesis) method Principal Component Analysis [LiJa88], to select a different number of features. However, no improvement in the classification performance is observed. We will further investigate these approaches in future studies.

Results Analysis

From Table 6.3.1, we can see that the classification along the *Focus* dimension by the Maximum Entropy classifier yields an average *F-measure* of 0.70, which suggests that machine learning methods can perform the *Focus* classification with good accuracy.

The investigation into the misclassification does not suggest a clearly effective way to further improve the *Focus* classification on the current dataset. A few misclassifications are caused by the removal of discriminating terms in the preprocessing step since they only occur once. For example, cue phrases such as "*not fully understand*", "*studies are limited*" for the *Generic* category, and "*We developed*" for the *Methodology* category are all filtered out. However, most of the misclassifications are caused by the fact that the available terms in the fragment representation cannot support a clear category assignment, especially for the *Methodology* and *Generic* categories.

Since we divide the *Focus* of a sentence into three categories: *Scientific*, *Methodology*, and *Generic*, the coarse granularity of the category definition allows a wide range of subjects for each category. In contrast to the classification along the other four dimensions, there is no fixed set of rules (e.g. explicit citations for *Evidence*) or cue terms (e.g. speculative words for *Certainty*) available to clearly define the category boundaries along the *Focus* dimension. To derive the general characteristics of each category (e.g., the frequently occurring words or phrases), a large amount of training data is necessary. In our dataset, only about 30 fragments are available for each of the *Methodology* and *Generic* categories. Such a limited number of examples cannot provide enough information to characterize each category. Since the category distribution is skewed, the classification decisions are then biased toward the major category, *Scientific*, as can be seen from the confusion matrix in Table 6.3.3. We expect that a larger amount of data with a relatively uniform category distribution can help to improve the classification performance along this dimension in the future.

Table 6.3.3. The confusion matrix of the Focus classification by the Maximum Entropy classifier.

	S	G	M
S	269.5	4.0	3.0
G	27.0	9.0	1.0
M	14.5	0.0	26.0

Compared to the classification on the *Focus* of a fragment, the classification on its *Evidence* level yields a better performance as shown in Table 6.3.2. The *F-measure* above

0.8 obtained by each classifier indicates that the categories can be recognized with high accuracy. To analyze the classification results, we start from the confusion matrix shown in Table 6.3.4. From the confusion matrix, we learn that there are mainly two kinds of misclassifications: the misclassification of category *E3* as *E0*, and the confusion between *E1* and other categories. We next examine them more closely.

Table 6.3.4. The confusion matrix of the Evidence classification by the Maximum Entropy classifier.

	E0	E1	E2	E3
E0	116	1	2	2
E1	10	11	1	5
E2	7	0	81	4
E3	20	1	2	91

There are two major causes for the misclassification of fragments under category *E3* into category *E0*.

First, many distinguishing terms for *Explicit evidence* (*E3*) such as "*our results reveal*", "*we report*", "*we provide evidence*" only occur once in the training set, and they are thereby filtered out automatically. Currently, we do not have an effective solution to this problem, since the occurrence of such terms must be significant in the training set to be recognized as cue terms for the category (*E3*). We expect that a relatively consistent set of phrases that are commonly used to express *Explicit evidence* in scientific literature, such as "*our data show*", "*our results indicates*", "*we found …*", "*we see …*", etc, can be derived on a sizable training corpus. This assumption is yet to be investigated in future studies.

Second, the *Evidence* level of some fragments is mainly decided by the semantic meaning of the whole statement, instead of a fixed set of words or phrases. For example, when a certain methodology of a scientific experiment is discussed, the *Evidence* level is typically *E3*; when the general state of knowledge is introduced, the *Evidence* level is usually *E0*. In such a case, it is hard to distinguish between *E0* and *E3* since no fixed terms are available to characterize each category. However, as we have discussed in Section 5.3.3, there may exist strong correlation between categories from the *Focus* and *Evidence* dimensions, such as *M* and *E3*, i.e. *Methodology* and *Explicit evidence*. If we can improve the prediction accuracy for the *Focus* classification when more examples are available, and make use of such correlation, the misclassification between *E0* and *E3* may be reduced.

To verify that correlation exists between the *Focus* and *Evidence* dimensions, we first revisit the feature definition in Section 5.3.3. For each fragment, we introduce a 3×4 feature matrix f^R (equation 5.36) to represent the co-occurrence between categories from the *Fragment Focus* and *Fragment Evidence* category spaces. Accordingly, we introduce a 3×4 parameter matrix λ^R to represent the weight of the feature matrix f^R in the conditional probability distribution $p(c \mid d)$ (equation 5.40). Table 6.3.5 shows the value of λ^R estimated over the training set, where the rows correspond to the *Focus* categories, and the columns correspond to the *Evidence* categories.

We can see that the parameter value reflects the likelihood of each possible combination between the categories along the two dimensions. We do not compare the absolute element values in Table 6.3.5. Since there are many more examples from the *Scientific* category than those from the *Methodology* or *Generic* category, the likelihood of each *Evidence* level to be combined with a *Scientific Focus* is high, compared to its combination with the other two categories. The comparison between the element values within the same row reveals some useful statistics. If the *Focus* of an example is *Scientific*, the possibility that it belongs to every *Evidence* category is similar, except for *E0* that is relatively higher. However, if an example is from the *Generic* category, it is highly likely that its *Evidence* level is *E0*; if an example is under the *Methodology* category, it is very probable that it has the highest *Evidence* level, i.e., *E3*. Therefore, if we can in the future improve the prediction accuracy for the *Generic* and *Methodology* categories, the misclassification between *E0* and *E3* may be reduced.

Table 6.3.5. The parameter value of λ^R.

	E0	E1	E2	E3
S	1.839	0.898	0.981	1.047
G	**1.110**	0.064	0.306	0.202
M	0.008	0.058	0.023	**0.813**

The confusion between *E1* and the other categories accounts for another source of misclassification. Unlike *E2* and *E3,* there are no predefined rules (e.g., reference to citations or figures) that can explicitly characterize the category *E0*. The classification is typically decided by the presence of certain terms, such as "*as previously discussed*", or "*studies are limited*". This accordingly requires a large number of training examples such that a set of commonly used cue terms may be derived. In our dataset, the examples under *E1* are much fewer than those under the other three categories (Figure 6.3.2), thereby, there is less information regarding this category. By inspecting the misclassified examples, we have learned that the removal of low-frequency cue words or phrases results in most of the misclassifications for the category *E1*. This kind of misclassification is hard to avoid, since the expressions for the indirect implication of evidence are very diverse. Typically with the inclusion of more examples, new, less-frequent cue terms are introduced.

Another issue that has caused the misclassification between *E1* and other categories is that the presence of certain predictive terms can support several levels of evidence. For example, the presence of the word “*indicate*” or “*suggest*” alone may imply the *Evidence* level of *E1*. However, when it co-occurs with external citations, the *Evidence* level may be *E2*. While when it is used together with a figure or a table, or as a part of a phrase like “*these findings indicate that*”, the *Evidence* level becomes *E3*. This kind of ambiguity can easily lead to misclassification on our limited training dataset, where the co-occurrence of different cue terms does not produce statistically significant patterns to distinguish different categories. This issue may be improved on a larger dataset and remains to be studied.

Overall, the preliminary results indicate that accurate *Evidence* classification can be obtained using machine learning algorithms. From the above analysis, we believe that certain issues regarding the misclassification may be improved when more data examples are available. The classification performance on a larger dataset remains to be investigated in the future.

Further Discussion

In our current dataset, there are several problems that may restrict the performance of the Maximum Entropy classifier.

First, the number of training examples is limited and the data representation is sparse. As previously mentioned, the highest prediction accuracy is obtained when 30 terms are included in the fragment representation for the *Focus* dimension. As a result, the data representation is very sparse and most entries in the fragment representation vector get a value of 0. Such a small and sparse dataset typically leads to poor classification performance.

Second, one advantage of our model is its ability to capture the correlation between categories in multi-label classification. However, the fraction of examples with multiple categories is very small in our dataset (Figure 6.3.1). As a consequence, useful statistics regarding the category correlation cannot be derived from the training data to help make classification decisions.

Last, an important feature of our model is that it takes into account the surrounding context when determining the *Focus* of a fragment. However, as previously mentioned, most sentences are not fragmented in our dataset. For an unfragmented sentence, the information from the sentence representation is the same as that from the fragment representation. Moreover, as previously mentioned, the optimal performance is obtained when only 30 terms are selected for the *Focus* classification. In such a case, both the fragment and sentence representations are sparse. As a result, a sentence usually contains no more information than the individual fragments within it. Considering the above factors, the merit brought by considering the context from the whole sentence is minor. Instead, the performance may degrade due to the addition of new features related to the surrounding context, since a model with a large number of parameters is more prone to overfitting on limited training examples. To address this issue, we modified the Maximum Entropy model to remove the constraints regarding context information. The experiment shows that the modified model yields a better classification performance than the original one (Table 6.3.6), specifically, the *F-measure* along the *Focus* dimension increases from 0.70 to 0.74.

Table 6.3.6. The classification performance along the Focus and Evidence dimension without considering context information in the Maximum Entropy model.

Category	Precision	Recall	F-measure	Accuracy
S	0.86	0.99	0.92	0.87
G	1.00	0.22	0.36	
M	0.91	0.63	0.75	
Average	**0.92**	**0.61**	**0.74**	
Average (Model with Context)	0.68	0.65	0.70	0.86

Category	Precision	Recall	F-measure	Accuracy
E0	0.77	0.96	0.85	0.85
E1	0.85	0.41	0.55	
E2	0.94	0.87	0.90	
E3	0.89	0.82	0.85	
Average	**0.86**	**0.76**	**0.81**	
Average (Model with Context)	0.86	0.76	0.81	0.85

In future studies, we will further examine the performance of our model on a sizable dataset which contains more sentences with multiple fragments and more fragments with multiple categories.

6.3.2 Certainty

The classification performance along the *Certainty* dimension is evaluated on the dataset *Frag_C*, which contains 879 sentence fragments. Before presenting the performance result, we first examine the distribution of fragments in the categories. Figure 6.3.4 demonstrates a very skewed distribution: among the 879 fragments, only about 60 are from the categories *Complete uncertainty*, *Low certainty*, or *High likelihood*.

As discussed in Section 4.1, each sentence fragment is mapped to a vector of words and *n*-grams in the preprocessing step. Two classifiers, Naïve Bayes and SVM, are applied to the dataset. The best performance is obtained when the top 100 terms returned by the *chi-square* term selection function are used to train the classifiers. Table 6.3.7 shows the performance measured in terms of *Precision*, *Recall* and *F-measure*.

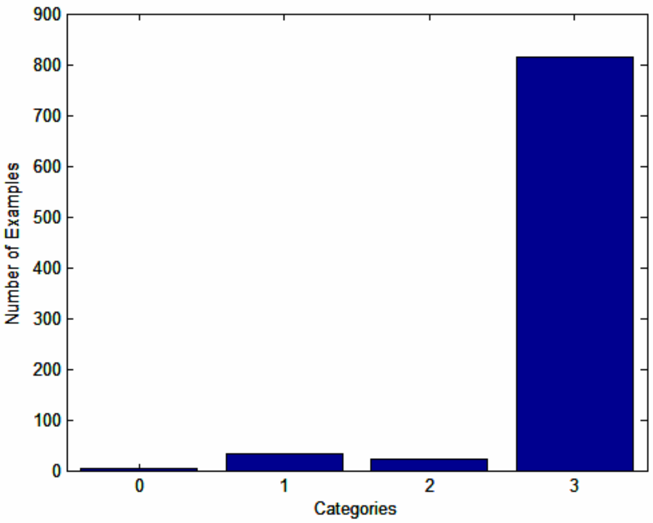


Figure 6.3.4. The distribution of fragments in the categories along the Certainty dimension over the dataset Frag_C. 0 denotes ***Complete uncertainty****, 1 denotes* ***Low certainty****, 2 denotes* ***High likelihood****, and 3 denotes* ***Complete certainty****.*

Table 6.3.7. The performance of the Certainty classification on the dataset Frag_C. The dataset contains 796 sentences (876 fragments). 0 denotes ***Complete uncertainty****, 1 denotes* ***Low certainty****, 2 denotes* ***High likelihood****, and 3 denotes* ***Complete certainty****.*

Category	Precision		Recall		F-measure		Accuracy	
	Naïve Bayes	SVM	Naïve Bayes	SVM	Naïve Bayes	SVM	Naïve Bayes	SVM
0	0.00	0.00	0.00	0.00	0.00	0.00	0.95	0.95
1	0.86	0.90	0.58	0.55	0.69	0.68		
2	0.96	0.67	0.13	0.09	0.23	0.15		
3	0.86	0.95	1.00	1.00	0.92	0.98		
Average	**0.67**	0.63	**0.43**	0.41	**0.52**	0.50		

The major issue regarding the classification along the *Certainty* dimension is that the distribution of fragments in the categories is strongly biased. According to the Annotation Guidelines [ShWR06], the rule for assigning *Complete certainty* is that no explicit indication of any degree of *uncertainty* is present in the fragment. Since we have only about 60 fragments belonging to the three categories of *uncertainty*, the examples that can be used to characterize the categories are about 60 despite almost 900 fragments available in the dataset. Furthermore, there exists some inconsistency in the annotated examples. For example, fragments containing the words “*indicating*”, “*suggesting*”, and “*possible*” are assigned to different degrees of *Certainty* by different groups of annotators. Consequently, the discriminating power of such words degrades and the misclassification caused by the annotation conflicts cannot be avoided. The lack of examples and the annotation conflicts result in a relatively low performance along this dimension as shown in Table 6.3.7. However, we note that this performance (an average *F-measure* of 0.50 or 0.52 in the above table) is much better than that obtained by a simple baseline classifier, which would, on such a highly skewed dataset, assign all the examples to the majority category and would yield an average *F-measure* of 0.24.

By inspecting the top ranking terms, we found that a set of cue terms such as "*may*", "*might*", "*can*", "*likely*", "*presumably*", "*poorly*", " *could be argued*", etc., are successfully detected. Since expressions of uncertainty are relatively consistent among scientific articles, one possible approach to improve the prediction accuracy is to manually include the most frequently used cue terms in a thesaurus. The thesaurus can be used as an artificial data source to pre-train the classifier, or post-process the classified examples. Thus, even if a predefined cue word rarely occurs in the training set, it can still be recognized as conveying a certain level of uncertainty. With the predefined thesaurus as an additional source of information, the classification performance may be further improved.

6.3.3 Polarity

The classification along the *Polarity* dimension is relatively easy compared to other dimensions, since the set of words that are typically used to express negation in scientific articles is small. The dataset we used for the *Polarity* classification, *Frag_P*, contains 1031 sentence fragments. All the fragments are mapped to vectors of words and syntactical phrases, which are classified by an SVM classifier. Figure 6.3.5 and Table 6.3.8 show the distribution of fragments in the categories and the classification result.

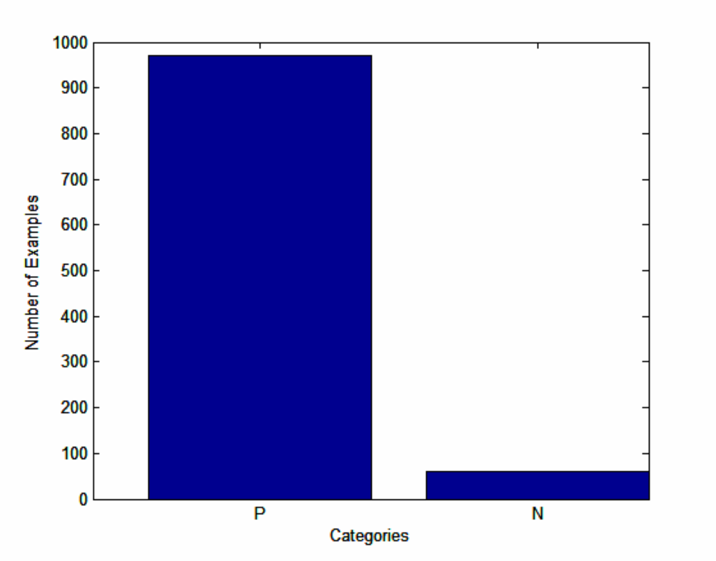


Figure 6.3.5. The distribution of fragments in the categories along the Polarity dimension over the dataset Frag_P. P denotes ***Positive****, and N denotes* ***Negative****.*

Table 6.3.8. The performance of the Polarity classification on the dataset Frag_P. The dataset contains 916 sentences (1031 fragments). P denotes ***Positive****, and N denotes* ***Negative****.*

Category	Precision	Recall	F-measure	Accuracy
P	0.99	0.99	0.99	0.99
N	0.86	0.89	0.87	
Average	**0.93**	**0.94**	**0.93**	

We can see that the prediction accuracy is high for the *Polarity* classification. Only a few examples are misclassified. Since some negative expressions such as "*neither*", "*nor*", "*none*" occur only once in the training set, the associated fragments are misclassified as *Positive* examples. Several misclassifications are caused by the failure to recognize the word "*not*" as a part of phrases such as "*whether or not*", "(*data not shown)*", because they are not considered as syntactical phrases. To address this issue, we investigate the representation consisting of words and statistical phrases, and the performance is slightly improved. Statistical phrases can capture more co-occurrence patterns of words which cannot be recognized by syntactical phrases. However, along with the introduction of more phrases, more redundant and noisy features may be included.

6.3.4 Trend

The classification performance along the *Trend* dimension is evaluated on the dataset *Frag_T*, where 822 sentences are broken into 919 fragments. Each fragment is represented as a vector of words and syntactical phrases. We first examine the distribution of fragments among categories (Figure 6.3.6). As can be seen from Figure 6.3.6, the distribution of fragments among categories is strongly biased along the *Trend* dimension. Although we have 919 fragments in the training dataset, fewer than 60 belong to the categories *Increase* or *Decrease*. As a result, the classification performance is not as good as along some of the other dimensions, as shown in Table 6.3.9. Again, we note that the result achieved here (an average *F-measure* of 0.72) is still much better than that obtained by simply classifying all the examples into the majority category (which would yield an average *F-measure* of 0.32).

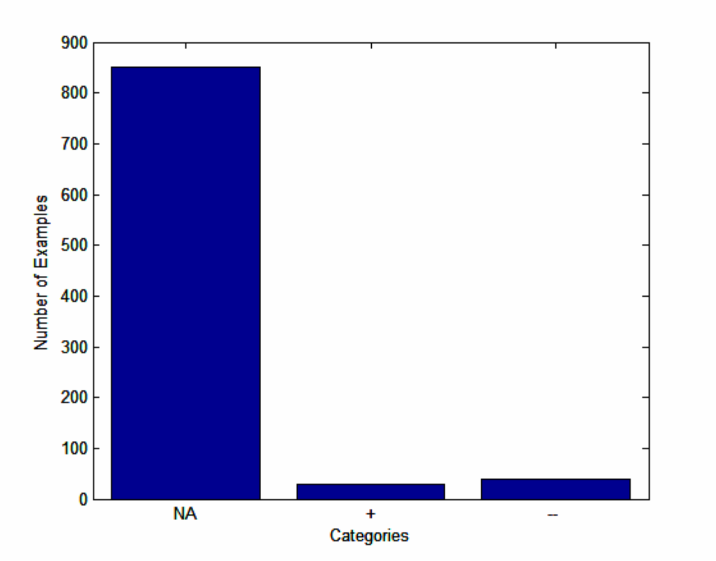


Figure 6.3.6. The distribution of fragments in the categories along the Trend dimension over the dataset Frag_T. NA denotes ***No trend,*** *+ denotes* ***Increase****, and – denotes* ***Decrease****.*

Table 6.3.9. The performance of the Trend classification on the dataset Frag_T. The dataset contains 822 sentences (919 fragments). NA denotes ***No trend,*** *+ denotes* ***Increase****, and – denotes* ***Decrease****.*

Category	Precision	Recall	F-measure	Accuracy
None	0.96	0.99	0.97	0.95
+	0.75	0.64	0.69	
-	0.72	0.33	0.46	
Average	**0.81**	**0.65**	**0.72**	

The major part of the misclassification for the categories *Increase* and *Decrease* (around 70 percent) is due to the inconsistency in annotation among different groups. For example, the word "*activation*" is typically considered to indicate an *Increase* direction by the first group of annotators (the guideline authors), while it is ignored by other groups of annotators. Similarly, the word "*reduce*", "*prevent*", or "*suppress*" is considered as an implication of a *Decrease* trend by the authors of the guideline, but not by other annotators. As a consequence, these cue terms are discarded by the classifier. In fact, most of the examples labeled with an *Increase* or a *Decrease Trend* were annotated by the authors of the guideline. Further revision is necessary on the annotation along the *Trend* dimension. About 15 percent of the misclassification is caused by the filtering out of less-frequent cue terms.

Based on the observation of the classification result, we found that terms such as "*increase*", "*decrease*", "*inhibit*", "*diminish*", etc., can be recognized as distinguishing terms. More importantly, we have found that such cue terms occur with high frequency in the small training set, which means that the set of words commonly used to express an *Increase* or a *Decrease* trend in biomedical literature is relatively fixed. We expect that performance can be further improved when we solve the annotation inconsistency problem and add in more data. Furthermore, the approach of building a thesaurus to include the most frequently used cue terms can also be considered.

6.4 Further Investigation on Larger Datasets

As previously mentioned, a major issue at the current stage is the lack of training examples. At the beginning of this chapter, we introduced four larger datasets generated based on the majority annotation agreement (Table 6.1.2). As further investigation, we experiment with the Maximum Entropy model along the *Focus* and *Evidence* dimensions, and SVM classifiers along other dimensions over these datasets. The distribution of fragments among categories along each dimension are shown in Figures 6.4.1 – 6.4.5. The classification performance, as well as the comparison with previous results over the datasets based on unanimous annotation agreement, are presented in Tables 6.4.1 – 6.4.5.

The experimental results show that the performance along the *Focus* and *Certainty* dimensions improves on the larger datasets. The addition of more examples leads to a less biased distribution of fragments among categories along the *Focus* dimension (Figure 6.4.1 vs. Figure 6.3.1). As a result, the *Focus* classification is further improved as we expected. The *F-measure* along the *Certainty* dimension remains low. The number of fragments belonging to categories of *uncertainty* is still small (around 200), and the cue phrases are very diverse. To learn a set of commonly used cue terms that can characterize each *uncertainty* category, we still need more examples.

The performance along the *Polarity* dimension remains similar, which implies the prediction accuracy along this dimension is relatively stable. The performance along the *Evidence* and *Trend* dimensions degrades. As previously mentioned in Section 6.3.4, the inconsistent annotation along the *Trend* dimension may account for part of the misclassifications with the addition of new data. Before solving this issue, it is hard to obtain high accuracy for the *Trend* classification. The variation in the performance along the *Evidence* dimension is not surprising. With the addition of new data, many less-frequent cue terms are introduced (especially for the category *E1* which is not characterized by a limited set of cue terms), and the removal of such terms inevitably leads to misclassification.

Most important, the annotations are not agreed upon by all three annotators, which may have introduced more inconsistencies into the datasets. The inferior data quality may result in less accurate classification. Further investigation and possible solutions to improve the classification performance are left for a future study.

Overall, the classification performance along the *Focus*, *Evidence* and *Polarity* dimensions remains good on the datasets generated by the majority annotation agreement, yielding an *F-measure* of 0.75, 0.74, and 0.93 respectively.

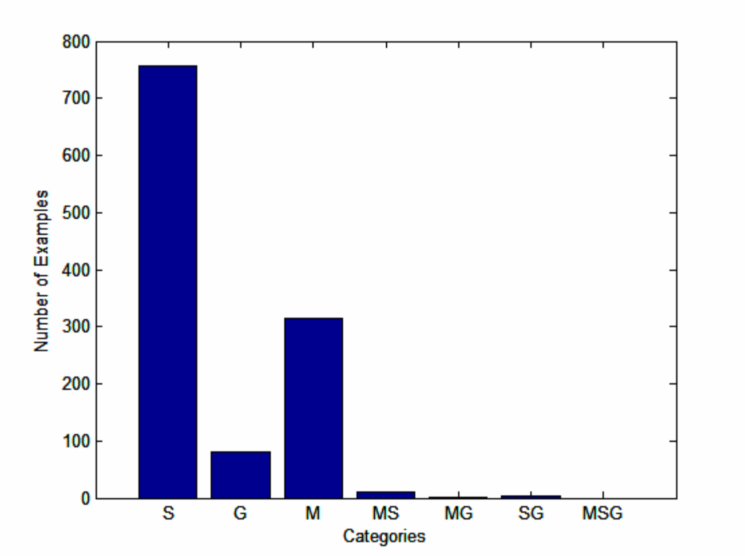


Figure 6.4.1. The distribution of fragments in the categories along the Focus dimension over the dataset Frag_M_FE. S denotes ***Scientific****, G denotes* ***Generic****, M denotes* ***Methodology****, MS, MG, SG, and MSG denote their combinations.*

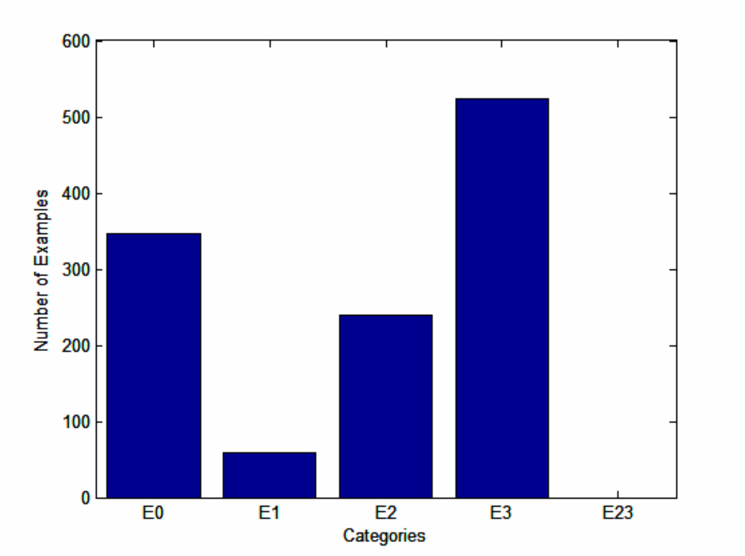


Figure 6.4.2. The distribution of fragments in the categories along the Evidence dimension over the dataset Frag_M_FE. E0 denotes ***No evidence****, E1 denotes* ***Claim of evidence without verifying information****, E2 denotes* ***Explicit citation****, E3 denotes* ***Explicit evidence****, and E23 denotes both* ***Explicit citation*** *and* ***Explicit evidence****.*

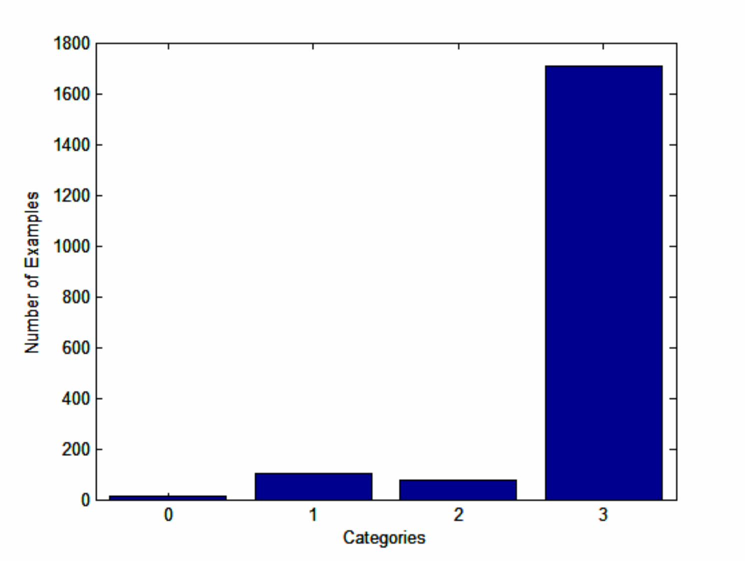


Figure 6.4.3. The distribution of fragments in the categories along the Certainty dimension over the dataset Frag_M_C. 0 denotes ***Complete uncertainty****, 1 denotes* ***Low certainty****, 2 denotes* ***High likelihood****, and 3 denotes* ***Complete certainty****.*

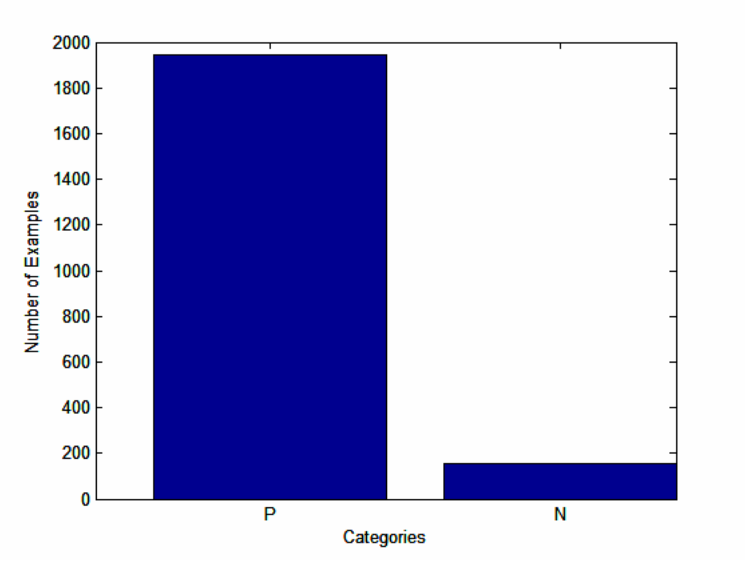


Figure 6.4.4. The distribution of fragments in the categories along the Polarity dimension over the dataset Frag_M_P. P denotes ***Positive****, and N denotes* ***Negative****.*

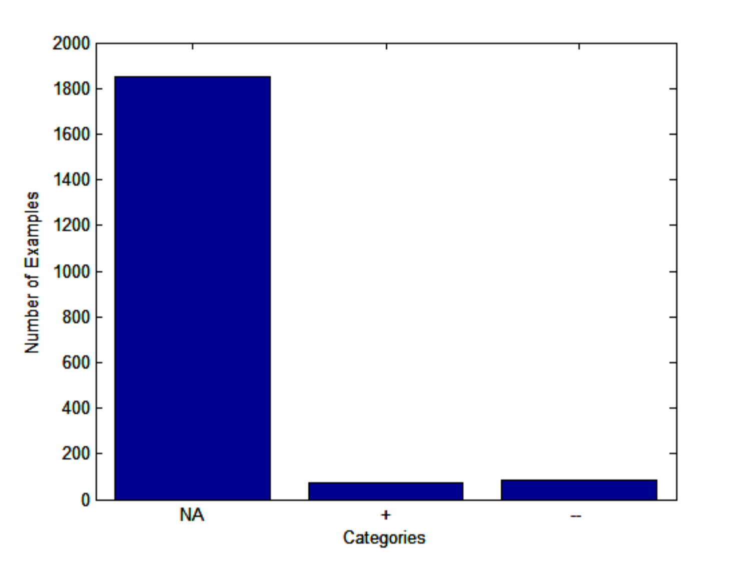


*Figure 6.4.5. The distribution of fragments in the categories along the Trend dimension over the dataset Frag_M_T. NA denotes **No trend,** + denotes **Increase**, and – denotes **Decrease**.*

*Table 6.4.1. The performance of the Focus classification on the dataset Frag_M_FE. The dataset contains 1051 sentences (1168 fragments). S denotes **Scientific**, G denotes **Generic**, and M denotes **Methodology**.*

Category	Precision	Recall	F-measure	Accuracy
S	0.88	0.94	0.91	0.86
G	0.73	0.34	0.47	
M	0.81	0.81	0.81	
Average	**0.81**	**0.70**	**0.75**	
Average (unanimous agreement)	0.81	0.62	0.70	0.86

*Table 6.4.2. The performance of the Evidence classification on the dataset Frag_M_FE. The dataset contains 1051 sentences (1168 fragments). E0 denotes **No evidence**,* E1 *denotes **Claim of evidence without verifying information**, E2 denotes **Explicit citation**, and E3 denotes **Explicit evidence**.*

Category	Precision	Recall	F-measure	Accuracy
E0	0.70	0.83	0.76	0.79
E1	0.71	0.43	0.54	
E2	0.81	0.73	0.77	
E3	0.86	0.82	0.84	
Average	**0.77**	**0.70**	**0.74**	
Average (unanimous agreement)	0.86	0.76	0.81	0.85

*Table 6.4.3. The performance of the Certainty classification on the dataset Frag_M_C. The dataset contains 1671 sentences (1902 fragments). 0 denotes **Complete uncertainty**, 1 denotes **Low certainty**, 2 denotes **High likelihood**, and 3 denotes **Complete certainty**.*

Category	Precision	Recall	F-measure	Accuracy
0	0.75	0.20	0.32	0.92
1	0.71	0.52	0.60	
2	0.33	0.18	0.23	
3	0.95	0.99	0.97	
Average	**0.69**	**0.47**	**0.56**	
Average (unanimous agreement)	0.63	0.41	0.50	0.95

*Table 6.4.4. The performance of the Polarity classification on the dataset Frag_M_P. The dataset contains 1819 sentences (2100 fragments). P denotes **Positive**, and N denotes **Negative**.*

Category	Precision	Recall	F-measure	Accuracy
P	0.99	0.99	0.99	0.98
N	0.88	0.86	0.87	
Average	**0.93**	**0.92**	**0.93**	
Average (unanimous agreement)	0.93	0.94	0.93	0.99

*Table 6.4.5. The performance of the Trend classification on the dataset Frag_M_T. The dataset contains 1739 sentences (2003 fragments). NA denotes **No trend,** + denotes **Increase**, and – denotes **Decrease**.*

Category	Precision	Recall	F-measure	Accuracy
None	0.95	0.98	0.97	0.94
+	0.55	0.38	0.45	
-	0.74	0.35	0.48	
Average	**0.75**	**0.57**	**0.65**	
Average (unanimous agreement)	0.81	0.65	0.72	0.95

6.5 Conclusion

We have conducted fragment classification experiments along the five dimensions: *Focus*, *Evidence*, *Certainty*, *Polarity*, and *Trend*. Our results suggest that machine learning algorithms can perform the annotation task along certain dimensions with good accuracy. The major issue at the current stage is the lack of training examples, which leads to the loss of many cue terms that are critical in deciding category boundaries. Moreover, the annotation inconsistency along certain dimensions accounts for another source of misclassification. From the analysis of the experimental results, we believe that some problems regarding the misclassifications may be solved with the addition of more data. The classification performance on larger datasets remains to be studied in the future.

Chapter 7

Extensions and Future Directions

In our future work, we plan to investigate several extensions to the Maximum Entropy model, as well as other possible approaches that may improve the performance of the fragment annotation task.

7.1 Further Extension to the Maximum Entropy Model

We have designed a new, preliminary model which aims to address three distinct characteristics in the fragment classification along the *Focus* and *Evidence* dimensions. The classification performance of this model shows that fragments can be classified with high accuracy. However, there are two main drawbacks in the proposed model. First, as mentioned in Section 5.3, finding the best change in each parameter at each training iteration requires the enumeration of 245 possible class labels for a fragment. The explicit identification of the *Sentence Focus*, which is not our primary concern, dramatically increases the computational complexity[1] of the model. Second, the relative positions among fragments are ignored when a three-dimensional vector is used to represent the *Sentence Focus*. In practice, among all the fragments in a sentence, those closer to the target fragment should have greater influence on its topic. To address the above problems, we propose three alternative models.

[1] The number of possible class labels increases from 35 (7*5) to 245 (7*5*7). The model takes about two minutes to converge over a set of 354 fragments. If we remove constraints about the *Sentence Focus*, that is, examining 35 possible labels rather than 245, the model takes only a few seconds to converge. The PC configuration is as follows: AMD Athlon 2.21 GHz, 2GB RAM.

7.1.1 Alternative Models

We first remove the constraints pertaining to the *Sentence Focus*, namely, the constraints over the category distribution within the *Sentence Focus* category space:

$$E_{\tilde{p}}(f_{Ni}^{P}) = E_{p}(f_{Ni}^{P}),\ 1 \le i \le 3;$$

the constraints over the correlation between terms and categories within the *Sentence Focus* category space:

$$E_{\tilde{p}}(f_{Nij}^{T}) = E_{p}(f_{Nij}^{T}),\ 1 \le i \le 3, 1 \le j \le |t_N|;$$

the constraints over the correlation among categories within the *Sentence Focus* category space:

$$E_{\tilde{p}}(f_{Nij}^{M}) = E_{p}(f_{Nij}^{M}),\ 1 \le i < j \le 3;$$

and the constraints over the correlation between *Fragment Focus* and *Sentence Focus*:

$$E_{\tilde{p}}(f_{ij}^{N}) = E_{p}(f_{ij}^{N}),\ 1 \le i \le 3,\ 1 \le j \le 3.$$

Next, we introduce several new constraints to the alternative models. We discuss each model in detail in the following sections.

The First Alternative Model

In the first variation on the model, we introduce a fixed context window, which includes, for each target fragment, the set of terms pertaining to the *Focus* of its two preceding and two succeeding fragments in the same sentence. Accordingly, we add four new feature function matrices for each fragment-label pair (d, c), capturing the correlation between the *Focus* label of the target fragment and the terms pertaining to the *Focus* of the surrounding fragments, defined as:

$$f_F^{T_{-2}}(d,c) = c_F t_F^{d_{-2}},$$

$$f_F^{T_{-1}}(d,c) = c_F t_F^{d_{-1}},$$

$$f_F^{T_{+1}}(d,c) = c_F t_F^{d_{+1}},$$

$$f_F^{T_{+2}}(d,c) = c_F t_F^{d_{+2}},$$

where $t_F^{d_{-2}}, t_F^{d_{-1}}$ are the term vectors pertaining to the *Focus* of the two fragments before the target fragment; $t_F^{d_{+2}}, t_F^{d_{+1}}$ are the term vectors pertaining to the *Focus* of the two fragments after the target fragment. If any of the surrounding fragments does not exist in the target sentence, each element value in the corresponding term vector is set equal to 0.

To make the model produce the same expected values for the above features as derived from the training data, we add four new types of constraints:

$$E_{\tilde{p}}(f_F^{T_{-2}}) = E_p(f_F^{T_{-2}}),\ 1 \le i \le 3, 1 \le j \le |t_F|,$$

$$E_{\tilde{p}}(f_F^{T_{-1}}) = E_p(f_F^{T_{-1}}),\ 1 \le i \le 3, 1 \le j \le |t_F|,$$

$$E_{\tilde{p}}(f_F^{T_{+1}}) = E_p(f_F^{T_{+1}}),\ 1 \le i \le 3, 1 \le j \le |t_F|,$$

$$E_{\tilde{p}}(f_F^{T_{+2}}) = E_p(f_F^{T_{+2}}),\ 1 \le i \le 3, 1 \le j \le |t_F|,$$

where $|t_F|$ is the total number of terms related to the *Focus* of a fragment. Thus, the optimal probability distribution $p^*(c \mid d)$ has the following parametric form:

$$p^*(c \mid d) = \frac{1}{Z(d)} \exp(\lambda_F^P \cdot f_F^P + \lambda_E^P \cdot f_E^P + \lambda_F^M \cdot f_F^M + \lambda_E^M \cdot f_E^M + \lambda^R \cdot f^R + \lambda_F^T \cdot f_F^T$$

$$+ \lambda_E^T \cdot f_E^T + \lambda_F^{T_{-2}} \cdot f_F^{T_{-2}} + \lambda_F^{T_{-1}} \cdot f_F^{T_{-1}} + \lambda_F^{T_{+1}} \cdot f_F^{T_{+1}} + \lambda_F^{T_{+2}} \cdot f_F^{T_{+2}}),$$

where $\lambda_F^{T_{-2}}, \lambda_F^{T_{-1}}, \lambda_F^{T_{+1}}, \lambda_F^{T_{+2}}$ are $3 \times |t_F|$ matrices which represent the weights of the feature matrices $f_F^{T_{-2}}, f_F^{T_{-1}}, f_F^{T_{+1}}, f_F^{T_{+2}}$ respectively.

In the other two alternative models, instead of using a fixed context window, we use a dynamic one. The context window covers two types of features, static and dynamic. Static features refer to the terms related to the *Focus* of the surrounding fragments; dynamic features refer to their *Focus* labels predicted by the classification model during the parsing process. We consider two parsing directions, forward and backward.

The Second Alternative Model

For the forward parsing, we use the terms pertaining to the *Focus* of the two preceding fragments, as well as their *Focus* labels, as context information of the target fragment. We define four new feature function matrices for each fragment-label pair (d,c). Two of them capture the correlation between the *Focus* label of the target fragment and the terms pertaining to the *Focus* of the two preceding fragments, defined as:

$$f_F^{T_{-2}}(d,c) = c_F t_F^{d_{-2}},$$

$$f_F^{T_{-1}}(d,c) = c_F t_F^{d_{-1}},$$

where $t_F^{d_{-2}}, t_F^{d_{-1}}$ are the term vectors pertaining to the *Focus* of the two fragments before the target fragment. The other two capture the correlation between the *Focus* labels of the target and each of the preceding fragments, defined as:

$$f_F^{M_{-2}}(d,c) = c_F c_F^{d_{-2}},$$

$$f_F^{M_{-1}}(d,c) = c_F c_F^{d_{-1}},$$

where $c_F^{d_{-2}}, c_F^{d_{-1}}$ are the *Focus* labels of the two fragments before the target fragment. For the training data, $c_F^{d_{-2}}, c_F^{d_{-1}}$ are the annotated class labels; for unknown test data, they are the labels dynamically predicted by the classification model. If any of the surrounding fragments does not exist in the target sentence, each element value in the corresponding term vector and label vector is set equal to 0.

To make the model produce the same expected values for the above features as derived from the training data, we add four new types of constraints:

$$E_{\tilde{p}}(f_F^{T_{-2}}) = E_p(f_F^{T_{-2}}), \qquad 1 \le i \le 3, 1 \le j \le |t_F|,$$

$$E_{\tilde{p}}(f_F^{T_{-1}}) = E_p(f_F^{T_{-1}}), \qquad 1 \le i \le 3, 1 \le j \le |t_F|,$$

$$E_{\tilde{p}}(f_F^{M_{-2}}) = E_p(f_F^{M_{-2}}), \qquad 1 \le i \le 3,\ 1 \le j \le 3,$$

$$E_{\tilde{p}}(f_F^{M_{-1}}) = E_p(f_F^{M_{-1}}), \qquad 1 \le i \le 3, 1 \le j \le 3,$$

where $|t_F|$ is the total number of terms related to the *Focus* of a fragment. The probability distribution $p^*(c\,|\,d)$ is defined as:

$$p^*(c\,|\,d) = \frac{1}{Z(d)} \exp(\lambda_F^P \cdot f_F^P + \lambda_E^P \cdot f_E^P + \lambda_F^M \cdot f_F^M + \lambda_E^M \cdot f_E^M + \lambda^R \cdot f^R + \lambda_F^T \cdot f_F^T$$

$$+ \lambda_E^T \cdot f_E^T + \lambda_F^{T_{-2}} \cdot f_F^{T_{-2}} + \lambda_F^{T_{-1}} \cdot f_F^{T_{-1}} + \lambda_F^{M_{-1}} \cdot f_F^{M_{-1}} + \lambda_F^{M_{-2}} \cdot f_F^{M_{-2}}),$$

where $\lambda_F^{M_{-1}}, \lambda_F^{M_{-2}}$ are 3×3matrices, representing the weights of the feature matrices $_{-1}$ u

and ; are

$$f_F^{T_{+2}}(d,c) = c_F t_F^{d_{+2}} \ ,$$

$$f_F^{T_{+1}}(d,c) = c_F t_F^{d_{+1}} \ ,$$

where $t_F^{d_{+2}}, t_F^{d_{+1}}$ are the term vectors pertaining to the *Focus* of the two fragments after the target fragment. The other two capture the correlation between the *Focus* labels of the target and each of the two fragments following it, defined as:

$$f_F^{M_{+2}}(d,c) = c_F c_F^{d_{+2}} \ ,$$

$$f_F^{M_{+1}}(d,c) = c_F c_F^{d_{+1}} \ ,$$

where $c_F^{d_{+2}}, c_F^{d_{+1}}$ are the *Focus* labels of the two fragments after the target fragment. For the training data, $c_F^{d_{+2}}, c_F^{d_{+1}}$ are the annotated class labels; for unknown test data, they are the labels dynamically predicted by the classification model. If any of the surrounding fragments does not exist in the target sentence, each element value in the corresponding term vector and label vector is set equal to 0.

To make the model produce the same expected values for the above features as derived from the training data, we add four new types of constraints:

$$E_{\tilde{p}}(f_F^{T_{+2}}) = E_p(f_F^{T_{+2}}), \qquad 1 \le i \le 3, 1 \le j \le |t_F|,$$

$$E_{\tilde{p}}(f_F^{T_{+1}}) = E_p(f_F^{T_{+1}}), \qquad 1 \le i \le 3, 1 \le j \le |t_F|,$$

$$E_{\tilde{p}}(f_F^{M_{+2}}) = E_p(f_F^{M_{+2}}) \ , \qquad 1 \le i \le 3,\ 1 \le j \le 3,$$

$$E_{\tilde{p}}(f_F^{M_{+1}}) = E_p(f_F^{M_{+1}}) \ , \qquad 1 \le i \le 3, 1 \le j \le 3,$$

where $|t_F|$ is the total number of terms related to the *Focus* of a fragment. The probability distribution $p^*(c \mid d)$ is defined as:

$$p^*(c \mid d) = \frac{1}{Z(d)} \exp(\lambda_F^P \cdot f_F^P + \lambda_E^P \cdot f_E^P + \lambda_F^M \cdot f_F^M + \lambda_E^M \cdot f_E^M + \lambda^R \cdot f^R + \lambda_F^T \cdot f_F^T$$

$$+ \lambda_E^T \cdot f_E^T + \lambda_F^{T_{+2}} \cdot f_F^{T_{+2}} + \lambda_F^{T_{+1}} \cdot f_F^{T_{+1}} + \lambda_F^{M_{+1}} \cdot f_F^{M_{+1}} + \lambda_F^{M_{+2}} \cdot f_F^{M_{+2}}),$$

where $\lambda_F^{M_{+1}}, \lambda_F^{M_{+2}}$ are 3×3matrices, representing the weights of the feature matrices $f_F^{M_{+1}}$ and $f_F^{M_{+2}}$; $\lambda_F^{T_{+2}}, \lambda_F^{T_{+1}}$ are $3 \times |t_F|$ matrices, representing the weights of the feature matrices $f_F^{T_{+2}}$ and $f_F^{T_{+1}}$.

With the three new models, for each fragment, we only need to examine 35 possible labels ($c \equiv (c_F, c_E)$) instead of 245 ($c \equiv (c_F, c_E, c_N)$). In addition, we take into account the distance between fragments when defining context. We need further empirical studies to examine whether the new model can yield higher prediction accuracy with lower computational complexity.

7.1.2 Other Suggested Improvements

Aside from the introduction of new constraints, other approaches can also be considered to further improve the classification model from different aspects:

Gaussian Smoothing

As can be seen from previous sections, the Maximum Entropy model is prone to overfitting due to the large amount of free parameters. To address this problem, a Gaussian prior for smoothing Maximum Entropy models was proposed by Chen and Rosenfeld [ChRo99]. Further work may be done to incorporate a Gaussian prior into the current Maximum Entropy model to avoid overfitting.

Parameter Estimation Algorithm

Currently we use the standard IIS algorithm for parameter estimation. Due to the large number of free parameters, and the large number of label combinations to be examined, the training process is relatively slow[2]. Previous studies show that the conjugate gradient approach and variable metric methods are much more efficient than the iterative scaling method in natural language processing (NLP) classification tasks [Malo02]. In our future work, we will try the limited memory variable metric algorithm [Malo02] to improve the convergence rate.

Negative Correlation

At present, the correlation between features and categories, and the correlation among categories, are captured by the product of the vector elements, such as, $c_{Fi}t_{Fj}$, and $c_{Fi}c_{Ej}$. This method only takes into account the positive-correlation, i.e. the case when the values of both elements are 1, while neglects the negative-correlation, i.e. the case when the values of both elements are 0. More sophisticated computation methods, such as XNOR, will be investigated in future studies to better capture the positive-, as well as negative-correlations between objects.

Feature-Specific Weighting

In our current classification model, different types of features, such as the terms pertaining to the *Fragment Focus* and the terms pertaining to the *Sentence Focus*, are treated equally. In future studies, we will seek to attribute a different weight to different

[2] The model takes about two minutes to converge over a set of 354 fragments. The PC configuration is as follows: AMD Athlon 2.21 GHz, 2GB RAM.

types of features. For example, terms pertaining to the *Focus* of the target fragment may outweigh those pertaining to the *Focus* of the surrounding context, that is, the *Sentence Focus.*

7.2 Future Directions in Automatic Fragment Annotation

In Section 7.1, we have discussed several possible extensions to the Maximum Entropy model for the *Focus* and *Evidence* dimensions. Other improvements to the general fragment annotation task can also be investigated in future studies. One possible approach is to make use of related work on the rhetorical analysis of text. In Section 2.3, we have surveyed the existing work on the rhetorical relations between clauses and sentences, such as *evidence*, *contrast*, *concession*, and *explanation*, as well as work on rhetorical roles such as *Background*, *Related Work*, *Solution and Method*, or *Result*. To annotate sentences in scientific articles, we first need to break sentences into fragments and then classify the fragments. Rhetorical relations between text units may be employed to automatically perform sentence fragmentation, and rhetorical roles can be used to improve the fragment classification. For example, if the rhetorical role of a fragment is *Background*, it is very likely that the *Focus* of the fragment is *Generic.* If the rhetorical role is *Solution and Method*, it is more likely that the *Focus* of the fragment is *Methodology*. Moreover, the *Evidence* level of fragments that discuss *Solution and Method* or *Result* tend to be *Explicit evidence.* However, the complexity of rhetorical analysis increases with the expansion of the categories, and the feasibility of automating it decreases. This direction remains yet to be explored.

Chapter 8

Conclusion

With the increasing availability of biomedical publications, it becomes ever more challenging to locate valuable and reliable information within the large amount of text. To identify and characterize text that satisfies certain types of information needs, Wilbur *et al.* [WiRS06, ShWR06] have proposed an annotation scheme to manually categorize a sentence fragment along five dimensions: *Focus*, *Polarity*, *Certainty*, *Evidence*, and *Trend*. *Focus* distinguishes whether the text describes scientific facts, experimental methodology, or general state of knowledge. *Polarity* checks whether an assertion is stated positively or negatively. *Certainty* measures the degree of confidence regarding the validity of an assertion. *Evidence* specifies the strength of evidence with respect to a statement. *Trend* indicates whether an increase or a decrease in a specific phenomenon is reported.

Following the annotation scheme, the work presented here examined the feasibility and reliability of the automatic categorization of biomedical text along the five dimensions using machine learning techniques. We conducted experiments using a set of manually annotated sentences that were sampled from different sections of biomedical journal articles. A classification model based on Maximum Entropy, designed specifically for this purpose, as well as two other popular algorithms in the area of text categorization, Naïve Bayes and Support Vector Machine (SVM), were trained and evaluated on the manually annotated dataset. The results show that the performance along the *Focus*, *Polarity*, and *Evidence* dimensions seems promising, yielding an *F-measure* of 0.70, 0.81, and 0.93 respectively. The performance along the *Certainty* and *Trend* dimensions is relatively low, partly due to the lack of training data and the inconsistent annotation. Compared to other biomedical text categorization tasks, which typically yielded an average *F-measure* between 0.4 to 0.7 [KDDC02, TRGN05], our classification task is well-defined and the results achieved are promising. We thereby conclude that machine learning methods can classify biomedical text fragments along multiple dimensions with good accuracy.

As future extensions, we suggested several approaches to further improve the classification model, as well as the performance of the general fragment annotation task. In our future work, we will examine the fragment classification on larger datasets. Moreover, we will extend the work to automatically process raw documents, that is, automatically breaking sentences into fragments, and using our classifier to annotate each fragment according to the predefined criteria.

As surveyed in Section 3.3, there are several applications for this work. These include: automated abstract generation (automatic text summarization), web page ranking, document categorization, named entity and relation extraction, and others. We believe that the annotation of text along multiple dimensions, which are defined to characterize several types of information needs, can lead to more accurate extraction and retrieval of information from a large volume of publications. We expect the automated annotation of text with reliable accuracy in the future can serve a variety of applications in the research community.

Bibliography

[Apos69] T. Apostol. **Calculus, volume II**. New York: John Wiley and Sons. pp. 314-318. 1969.

[BaMc98] L. Baker and A. McCallum. **Distributional clustering of words for text classification**. *Proceedings of the 21st ACM International Conference on Research and Development in Information Retrieval (SIGIR-98)*, pp. 96-103. 1998.

[BAOV99] C. Blaschke, M. Andrade, C. Ouzounis, and A. Valencia. **Automatic extraction of biological information from scientific text: protein-protein interaction**. *AAAI Conference on Intelligent Systems in Molecular Biology*, pp. 60-67. 1999.

[BeAl04] R. Bekkerman and J. Allan. **Using bigrams in text categorization**. *CIIR (the Center for Intelligent Information Retrieval, University of Massachusetts), Technical Report IR-408*. 2004.

[BeDD96] A. Berger, S. Della Pietra, and V. Della Pietra. **A Maximum Entropy approach to natural language processing**. *Computational Linguistics* 22(1), pp. 39-71. 1996.

[Berg97] A. Berger. **The improved iterative scaling algorithm: A gentle introduction**. www.cs.cmu.edu/afs/~aberger/www/ps/scaling.ps. 1997.

[Bert99] D. Bertsekas. **Nonlinear Programming**. Belmont, MA: Athena Scientific. 1999.

[Berg00] A. Berger. **Convexity, maximum likelihood and all that**. *CMU Computer Science Department Technical Report*. 2000.

[BioC04] http://biocreative.sourceforge.net/. 2004.

[BiSW99] D. Bikel, R. Schwartz, and R. Weischedel. **An algorithm that learns what's in a name**. *Machine Learning*, 34(1), pp. 211-231. 1999.

[BrSC05] T. Brow, B. Settles, and M. Craven. **Classifying biomedical articles by making localized decisions**. *Proceedings of the 14th Text Retrieval Conference (TREC).* 2005.

[BSAG98] A. Borthwick, J. Sterling, E. Agichtein, and R. Grishman. **Exploiting diverse knowledge sources via Maximum Entropy in named entity**

recognition. *Proceedings of the 6th Workshop on Very Large Corpora*. 1998.

[CaHo04] L. Cai and T. Hofmann. **Hierarchical document categorization with Support Vector Machines**. *Proceedings of the 13th ACM Conference on Information and Knowledge Management*, pp. 78-87. 2004.

[CaMS01] M. Caropreso, S. Matwin, and F. Sebastiani. **A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization**. *Text Databases and Document Management: Theory and Practice*. Hershey, US: Idea Group Publishing. 2001.

[ChHM97] D. Chickering, D. Heckerman, and C. Meek. **A Bayesian approach for learning Bayesian networks with local structure**. *Proceedings of 13th Conference on Uncertainty in Artificial Intelligence,* pp. 80-89. 1997.

[ChLi01] C. Chang and C. Lin. **LIBSVM : a library for support vector machines**. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm. 2001.

[ChRo99] S. Chen and R. Rosenfeld. **A gaussian prior for smoothing Maximum Entropy models**. *CMU Computer Science Department Technical Report*. 1999.

[ChYa00] W. Chuang and J. Yang. **Extracting sentence segments for text summarization: A machine learning approach**. *Proceedings of the 23rd ACM International Conference on Research and Development in Information Retrieval (SIGIR-00)*, pp. 152-159. 2000.

[CoMS04] M. Couto, B. Martins, and J. Silva. **Classifying biomedical articles using web resources**. *Proceedings of the 2004 ACM Symposium on Applied Computing*, pp. 111-115. 2004.

[CoNT00] N. Collier, C. Nobata, and J. Tsujii. **Extracting the names of genes and gene products with a hidden Markov model**. *Proceedings of the 18th International Conference on Computational Linguistics (COLINC)*, pp. 201-207. 2000.

[CoTh91] T. Cover and J. Thomas. **Elements of Information Theory**. New York: John Wiley and Sons. 1991.

[CoWe03] C. Courcoubetis and R. Weber. **Pricing Communication Networks: Economics, Technology and Modeling**. Appendix A: Lagrangian Methods for Constrained Optimization. New York: John Wiley and Sons. 2003.

[CrKu99] M. Craven and J. Kumlien. **Constructing biological knowledge based by extracting information from text sources**. *Proceedings of the 7th*

International Conference on Intelligent Systems for Molecular Biology (ISMB), pp. 77-86. 1999.

[CrSi01] K. Crammer and Y. Singer. **On the algorithmic implementation of multi-class SVMs**. *Journal of Machine Learning Research*, 2, pp. 265-292. 2001.

[DBNW02] J. Ding, D. Berleant, D. Nettleton, and E. Wurtele. **Mining MEDLINE: abstracts, sentences, or phrases?** *Pacific Symposium on Biocomputing*, pp. 326-337. 2002.

[DeZa01] L. Denoyer and H. Zaragoza. **HMM-based passage models for document classification and ranking**. *Proceedings of ECIR-01, 23rd European Colloquium on Information Retrieval Research*, pp. 126-135. 2001.

[DPHS98] S. Dumais, J. Platt, D. Heckerman, and M. Sahami. **Inductive learning algorithms and representations for text categorization**. *Proceedings of CIKM-98, 7th ACM International Conference on Information and Knowledge Management*, pp. 148-155. 1998.

[DuCh00] S. Dumais and H. Chen. **Hierarchical classification of web content**. *Proceedings of the 23rd ACM International Conference on Research and Development in Information Retrieval (SIGIR-00)*, pp. 256-263. 2000.

[DuHa73] R. Duda and P. Hart. **Pattern Classification**. New York: John Wiley and Sons. 1973.

[Edmu68] H. P. Edmundson. **New methods in automatic extracting**. *Journal of the Association for Computing Machinery*, 16(2), pp. 264-285. 1968.

[EGJR05] F. Ehrler, A. Geissbuhler, A. Jimeno, and P. Ruch. **Data-poor categorization and passage retrieval for Gene Ontology annotation in Swiss-Prot**. *BMC Bioinformatics*, 6(Suppl 1). 2005.

[EsAg04] E. Eskin and E. Agichtein. **Combining text mining and sequence analysis to discover protein functional regions**. *Proceedings of the 9th Pacific Symposium on Biocomputing*, pp. 288-299. 2004.

[FAAC94] C. Friedman, P. Alderson, J. Austin, J. Cimino, and S. Johnson. **A general natural language text processor for clinical radiology**. *Journal of American Medical Informatics Association*, 1(2), pp. 161-174. 1994.

[Fuhr85] N. Fuhr. **A probabilistic model of dictionary-based automatic indexing**. *Proceedings of RIAO-85, 1st International Conference, Recherche d'Information Assistee par Ordinateur*, pp. 207-216. 1985.

[FuMR98] J. Furnkranz, T. Mitchell, and E. Riloff. **A case study in using linguistic phrases for text categorization on the WWW**. *Proceedings of the AAAI/ICML Workshop on Learning for Text Categorization*, pp. 5-12. 1998.

[Furn98] J. Furnkranz. **A study using *n*-gram features for text categorization**. *Technical Report OEFAI-TR-9830, Austrian Institute for Artificial Intelligence*. 1998.

[Gana02] M. K. Ganapathiraju. **Relevance of cluster size in MMR based summarizer: A report**. *Self-paced lab in Information Retrieval.* 2002.

[GKMC99] J. Goldstein, M. Kantrowitz, V. Mittal, and J.Carbonell. **Summarizing text documents: Sentence selection and evaluation metrics**. P*roceedings of the 22nd ACM International Conference on Research and Development in Information Retrieval (SIGIR-99)*, pp. 121-128. 1999.

[GO00] The Gene Ontology Consortium. **Gene Ontology: tool for the unification of biology**. *Nature Genet*, 25, pp. 25-29. 2000.

[HaRL05] J. Hakenberg, J. Rutsch, and U. Leser. **Tuning text classification for hereditary diseases with section weighting**. *The 1st International Symposium on Semantic Mining in Biomedicine (SMBM)*, pp. 34-39. 2005.

[HBBD06] A. Höglund, T. Blum, S. Brady, P. Dönnes, J. San Miguel, M. Rocherford, O. Kohlbacher and H. Shatkay. **Significantly improved prediction of subcellular localization by integrating text and protein sequence data**. *Proceedings of the Pacific Symposium on Biocomputing (PSB-06)*, pp. 16-27. 2006.

[HuDG00] K. Humphreys, G. Demetriou, and R. Gaizauskas. **Two applications of information extraction to biological science journal articles: Enzyme interactions and protein structures**. *Pacific Symposium on Biocomputing*. 2000.

[HuPS96] D. Hull, J. Pedersen, and H. Schutze. **Document routing as statistical classification**. *AAAI Spring Symposium on Machine Learning in Information Access Technical Papers*.1996.

[IbCT99] K. Ibushi, N. Collier, and J. Tsujii. **Classification of MEDLINE abstracts**. *Proceedings of Genome Informatics*. 1999.

[JJDv05] R. Jelier, G. Jenster, C. Dorssers, C. van der Eijk, M. van Mulligan, B. Mons, and J. Kors. **Co-occurrence based meta-analysis of scientific texts: retrieving biological relationships between genes**. *Bioinformatics*, 21(9), pp. 2049-2058. 2005.

[Joac98] T. Joachims. **Text categorization with Support Vector Machines: Learning with many relevant features**. *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pp. 137-142. 1998.

[Joac99] T. Joachims. **Transductive inference for text classification using support vector machines**. *Proceedings of ICML-99, 16th International Conference on Machine Learning*, pp. 200-209. 1999.

[Joac02] T. Joachims. **Optimizing search engines using clickthrough data**. *Proceedings of Knowledge Discovery in Databases*, pp. 133-142. 2002.

[KaMO04] Y. Kaneta, M. Munna, and T. Ohkawa. **A method of extracting sentences related to protein interaction from literature using a structure database**. *Proceedings of the 2nd European Workshop on Data Mining and Text Mining for Bioinformatics*, pp. 18-25. 2004.

[KDDC02] A. Yeh, L. Hirschman, and A. Morgan. **Background and overview for KDD Cup 2002 task 1: Information extraction from biomedical articles**. *SIGKDD Explorations: Newsletter of the ACM Special Interest Group on Knowledge Discovery and Data Mining*, 4, pp. 87-89. 2002.

[KiMF04] S. Kiritchenko, S.Matwin, and F. Famili. **Hierarchical text categorization as a tool of associating genes with Gene Ontology codes**. *The 2nd European Workshop on Data Mining and Text Mining for Bioinformatics held in conjunction with 15th European Conference on Machine Learning (ECML)*, pp. 26-30. 2004.

[KiMF05] S. Kiritchenko, S. Matwin, and F. Famili. **Functional annotation of genes using hierarchical text categorization**. *Proceedings of the BioLINK SIG: Linking Literature, Information and Knowledge for Biology*. 2005.

[Kiri05] S. Kiritchenko. **Hierarchical text categorization and its application to Bioinformatics**. PhD Thesis. 2005.

[KiWA97] J. Kivinen, M. Warmuth, and P. Auer. **The perceptron algorithm versus winnow: Linear versus logarithmic mistake bounds when few input variables are relevant**. *Artificial Intelligence*, 97(1-2), pp. 325-343. 1997.

[KnDa94] A. Knott and R. Dale. **Using linguistic phenomena to motivate a set of coherence relations**. *Discourse Processes*, 18, pp. 35-62. 1994.

[KnSa98] A. Knott and T. Sanders. **The classification of coherence relations and their linguistic markers: An exploration of two languages**. *Journal of Pragmatics*, 30, pp. 135-175. 1998.

[KoSa97] D. Koller and M. Sahami. **Hierarchically classifying documents using very few words**. *Proceedings of the 14th International Conference on Machine Learning*, pp. 170-178. 1997.

[KOSL02] S. Keerthi, C. Ong, K. Siah, D. Lim, W. Chu, and M. Shi. **A machine learning approach for the curation of biomedical literature - KDD Cup 2002 (Task 1)**. *SIGKDD Explorations: Newsletter of the ACM Special Interest Group on Knowledge Discovery and Data Mining*, 4, pp. 93-94. 2002.

[KuJM01] N. Kushmerick, E. Johnston, and S. McGuinness. **Information extraction by text classification**. *IJCAI-2001 Workshop on Adaptive Text Extraction and Mining*. 2001.

[KuMa00] T. Kudo and Y. Matsumoto. **Use of Support Vector learning for chunk identification**. *Proceedings of the 4th Conference on CoNLL-2000 and LLL-2000*, pp. 142-144. 2000.

[KuPC95] J. Kupiec, J. Pedersen and F. Chen. **A trainable document summarizer**. *Proceedings of the 18th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR-95)*, pp. 68-73. 1995.

[LaFi99] Y. Labrou and T. Finin. **Yahoo! as an ontology: Using Yahoo! categories to describe documents**. *Proceedings of the 8th International Conference on Information Knowledge Management*, pp. 180-187. 1999.

[LeCL04] J. Leonard, J. Colombe, and J. Levy. **Finding relevant references to genes and proteins in Medline using a Bayesian approach**. *Bioinformatics*, 18(11), pp. 1515-1522. 2004.

[LeCr90] D. Lewis and W. Croft. **Term clustering of syntactic phrases**. *Proceedings of the 13th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR-90)*, pp. 385-404. 1990.

[Leek97] T. Leek. **Information extraction using hidden Markov models**. MSc thesis, Dept. of Computer Science and Engineering, Univ. of California, San Diego. 1997.

[LeHR03] K. Lee, Y. Hwang, and H. Rim. **Two-phase biomedical NE recognition based on SVMs**. *Proceedings of ACL 2003 Workshop on Natural Language Processing in Biomedicine*, pp. 33-40. 2003.

[LeRi94] D. Lewis and M. Ringuette. **A comparison of two learning algorithms for text categorization**. *Proceedings of SDAIR-94, 3rd Annual*

Symposium on Document Analysis and Information Retrieval, pp. 81-93. 1994.

[Lewi92] D. Lewis. **An evaluation of phrasal and clustered representations on a text categorization task**. *Proceedings of the 15th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR-92)*, pp. 37-50. 1992.

[LiAD03] C. Liao, S. Alpha, and P. Dixon. **Feature preparation in text categorization**. *ADM03 workshop (The Australian Data Mining Workshop)*. 2003.

[LiBr04] M. Light and S. Bradshaw. **Annotating relations to events in Bioscience abstracts**. *BioLink 2004 Workshop on Linking Biological Literature, Ontologies and Databases: Tools for Users*. 2004.

[LiJa88] Y. Li and A. Jain. **Classification of text documents**. *The Computer Journal*, 41(8), pp. 537-546.1988.

[LiQS04] M. Light, X. Qiu, P. Srinivasan. **The Language of BioScience: Facts, speculations, and statements in between**. *Proceedings of BioLink 2004 Workshop on Linking Biological Literature, Ontologies and Databases: Tools for Users*. 2004.

[LTLS05] M. Lan , C. Tan , H. Low , and S. Sung. **A comprehensive comparative study on term weighting schemes for text categorization with support vector machines**. *Posters Proceedings 14th International World Wide Web Conference*, pp. 1032-1033. 2005.

[Luhn58] H. Luhn. **The automatic creation of literature abstracts**. *IBM Journal of Research & Development*, 2(2), pp. 159-165. 1958.

[MaEc02] D. Marcu and A. Echihabi. **An unsupervised approach to recognizing discourse relations**. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 368-375. 2002.

[Malo02] R. Malouf. **A comparison of algorithms for Maximum Entropy parameter estimation**. *Proceedings of the 6th Conference on Natural Language Learning (CoNLL-2002)*, pp. 49-55. 2002.

[Marc98] D. Marcu. **The rhetorical parsing, summarization, and generation of natural language texts**. PhD thesis, Department of Computer Science, University of Toronto. 1998.

[MaSM94] M. Marcus, B. Santorini, and M. Marcinkiewicz. **Building a large annotated corpus of English: the PennTreebank**. *Computational Linguistics*, 19, pp. 313-330. 1994.

[MaTh88] W. Mann and S. Thompson. **Rhetorical structure theory: Toward a functional theory of text organization**. *Text*, 8(3), pp. 243-281.1988.

[McFP00] A. McCallum, D. Freitag, and F. Pereira. **Maximum Entropy Markov models for information extraction and segmentation**. *Proceedings of the 17th International Conference on Machine Learning (ICML-2000)*, pp. 591-598. 2000.

[McNi98] A. McCallum and K. Nigam. **A comparison of event models for Naïve Bayes text classification**. *AAAI-98 Workshop on Learning for Text Categorization*. 1998.

[McSr03] L. McKnight and P. Srinivasan. **Categorization of sentence types in medical abstracts**. *Proceedings of the Annual Symposium of the American Medical Informatics Association (AMIA)*, pp. 440-444. 2003.

[MeDi03] R. Mercer and C. DiMarco. **The importance of fine-grained cue phrases in scientific citations**. *Proceedings of the 16th Conference of the CSCSI/SCEIO (AI-2003)*, pp. 550-556. 2003.

[MeDK04] R. Mercer, C. DiMarco, and F. Kroon. **The frequency of hedging cues in citation contexts in scientific writing**. *Proceedings of the 17th Conference of the CSCSI/SCEIO (AI-2004)*, pp. 75-88. 2004.

[MeSH06] Medical Subject Headings. http://www.nlm.nih.gov/mesh/introduction2006.html. 2006.

[MiCo04a] Y. Mizuta and N. Collier. **An annotation scheme for a rhetorical analysis of biology articles**. *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC2004)*, pp. 1737-1740. 2004.

[MiCo04b] Y. Mizuta and N. Collier, **Zone identification in biology articles as a basis for information extraction**. *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA, COLING 2004)*, pp. 29-35. 2004.

[Mill90] G. Miller. **WordNet: an on-line lexical database**. *International Journal of Lexicography*, 3(4), pp. 235-244. 1990.

[MlGr98] D. Mladenic and M. Grobelnik. **Word sequences as features in text learning**. *Proceedings of the 17th Electrotechnical and Computer Science Conference (ERK-98)*, pp. 145-148. 1998.

[Moul96] I. Moulinier. **A framework for comparing text categorization approaches**. *AAAI Spring Symposium on Machine Learning and Information Access*. 1996.

[MRMN98] A. McCallum, R. Rosenfeld, T. Mitchell, and A. Ng. **Improving text classification by shrinkage in a hierarchy of classes**. *Proceedings of the 15th International Conference On Machine Learning*, pp. 359-367. 1998.

[MuMC05] T. Mullen, Y. Mizuta, and N. Collier. **A baseline feature set for learning rhetorical zones using full articles in the biomedical domain**. *SIGKDD Explorations*, 7(1), pp. 52-58. 2005.

[Murt83] F. Murtagh. **A survey of recent advances in hierarchical clustering algorithms**. *The Computer Journal*, 26(4), pp. 354-359. 1983.

[NaRo02] R. Nair and B. Rost. **Inferring sub-cellular localization through automated lexical analysis**. *Bioinformatics*, 18 (Suppl 1), pp. S78-S86. 2002.

[NiLM99] K. Nigam, J. Lafferty, and A. McCallum. **Using Maximum Entropy for text classification**. *IJCAI-99 Workshop on Machine Learning for Information Filtering*, pp. 61-67, 1999.

[NSMU01] C. Nobata, S. Sekine, M. Murata, K. Uchimoto, M. Utiyama, and H. Isahara. **Sentence extraction system assembling multiple evidence**. *Proceedings of the 2nd NTCIR Workshop*, pp. 319-324. 2001.

[NSUI02] C. Nobata, S. Sekine, K. Uchimoto, and H. Isahara. **A summarization system with categorization of document sets**. *In Working Notes of the Third NTCIR Workshop Meeting*, pp. 33-38. 2002.

[OHTT01] T. Ono, H. Hishigaki, A. Tanigami, and T. Takagi. **Automated extraction of information on protein-protein interactions from the biological literature**. *Bioinformatics* 17, pp. 155-161. 2001.

[PNRH05] N. Polavarapu, S. Navathe, R. Ramnarayanan, A. Haque, S. Sahay, and Y. Liu. **Investigation into biomedical literature classification using Support Vector Machines**. *Proceedings of the IEEE Conference on Computational Systems in Bioinformatics*, pp. 366-374. 2005.

[Port80] M. Porter. **An algorithm for suffix stripping**. *Program*, 14(3), pp. 130-137. 1980.

[RaCr01] S. Ray and M. Craven. **Representing sentence structure in hidden Markov models for information extraction**. *Proceedings of the International Joint Conference on Artificial Intelligence*. 2001.

[RaRR94] A. Ratnaparkhi, J. Reymar, and S. Roukos. **A Maximum Entropy model for prepositional phrase attachment**. *Proceedings of the ARPA Human Language Technology Workshop)*, pp. 250-225. 1994.

[Ratn96] A. Ratnaparkhi. **A Maximum Entropy model for part-of-speech tagging**. *Proceedings of the 1st Conference on Empirical Methods in Natural Language Processing*, pp. 133-142. 1996.

[RCSA02] S. Raychaudhuri, J. Chang, P. Sutphin, and R. Altman. **Associating genes with gene ontology codes using a Maximum Entropy analysis of biomedical literature**. *Genome Research*, 12(1), pp. 203-214. 2002.

[RiNS05] B. Rice, G. Nenadic, and J. Stapley. **Mining protein function from text using term-based Support Vector Machines**. *BMC Bioinformatics*. 6 (Suppl 1). 2005.

[RJBT00] D. Radev, H. Jing, M. Budzikowska, and D. Tam. **Centroid-based summarization of multiple documents: Sentence extraction, utility-based evaluation, and user studies**. *ANLP/NAACL 2000 Workshop,* pp. 21-29. 2000.

[RuSr97] M. Ruiz and P. Srinivasan. **Automatic text categorization using neural networks**. *Proceedings of the 8th ASIS/SIGCR Workshop on Classification Research*, pp. 59-72. 1997.

[SaBu98] G. Salton and C. Buckley. **Term weighting approaches in automatic text retrieval**. *Information Processing and Management*. 24(5), pp. 513-523.1988.

[Saha96] M. Sahami. **Learning limited dependence Bayesian classifiers**. *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pp. 335-338. 1996.

[SaKi98] M. Sasaki and K. Kita. **Rule-based text categorization using hierarchical categories**. *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics,* pp. 2827-2830. 1998.

[SaSN92] T. Sanders, W. Spooren, and L. Noordman. **Toward a taxonomy of coherence relations**. *Discourse Processes*, 15(1), pp. 1-35. 1992.

[ScHP95] H. Schutze, D. Hull, and J. Pedersen. **A comparison of classifiers and document representations for the routing problem**. *Proceedings of the 18th ACM International Conference on Research and Development in Information Retrieval (SIGIR-95),* pp. 229-237. 1995.

[Seba99] F. Sebastiani. **Machine learning in automated text categorization**. *ACM Computing Surveys*, 34(1), pp. 1-47. 1999.

[SEWB00] H. Shatkay, S. Edwards, W. Wilbur, and M. Boguski. **Genes, themes and microarrays: using information retrieval for large-scale gene analysis**. *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pp. 317-328. 2000.

[ShWR06] H. Shatkay, W. Wilbur, and A. Rzhetsky. **Annotation Guidelines**. http://www.ncbi.nlm.nih.gov/CBBresearch/Wilbur/AnnotationGuidelines.pdf. 2006.

[SkCR03] M. Skounakis, M. Craven, and S. Ray. **Hierarchical hidden Markov models for information extraction**. *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI03)*, pp. 427-433. 2003.

[SmCl03] T. Smith and J. Cleary. **Automatically linking MEDLINE abstracts to the Gene Ontology**. *Proceedings of the ISMB 2003 BioLINK Text Data Mining*. 2003.

[SmRW04] L. Smith, T. Rindflesch and W. Wilbur. **MedPost: a part-of-speech tagger for bioMedical text**. *Bioinformatics*, 20(14), pp. 2320-2321. 2004.

[SuLN03] A. Sun, E. Lim, and W. Ng. **Performance measurement framework for hierarchical text classification**. *Journal of the American Society for Information Science and Technology (JASIST)*, 54(11), pp. 1014-1028. 2003.

[Taka04] H. Takamura. **Tutorial : Maximum Entropy model**. http://www.lr.pi.titech.ac.jp/~takamura/pubs/noteME.pdf. 2004.

[TaWi02] L. Tanabe and W. Wilbur. **Tagging gene and protein names in biomedical text**. *Bioinformatics*, 18(8), pp. 1124-1132. 2002.

[TeCM99] S. Teufel, J. Carletta, and M. Moens. **An annotation scheme for discourse-level argumentation in research articles**. *Proceedings of EACL*, pp. 110-117. 1999.

[TeMo97] S. Teufel and M. Moens. **Sentence extraction as a classification task**. *ACL/EACL-97 Workshop on Intelligent Scalable Text Summarization*, pp. 58-65. 1997.

[TeMo99] S. Teufel and M. Moens. **Argumentative classification of extracted sentences as a first step towards flexible abstracting**. *Advances in Automatic Text Summarization*, pp. 155-171. The MIT Press. 1999.

[TRFT02] TREC Filtering task. http://trec.nist.gov/. 1996-2002.

[TRGN05] TREC Genomics task. http://ir.ohsu.edu/genomics/. 2003-2005.

[VanR79] C. Van Rijsbergen. **Information Retrieval (Second ed.)**. London, UK: Butterworths. 1979.

[VeTS04] J. Vert, K. Tsuda and B. Schölkopf. **A primer on kernel methods**. *Kernel Methods in Computational Biology*, pp. 35-70. MIT Press. 2004.

[ViGi02] A. Vinokourov and M. Girolami. **A probabilistic framework for the hierarchic organization and classification of document collections**. *Journal of Intelligent Information Systems*, 18(2/3), pp. 153-172, Special Issue on Automated Text Categorization. 2002.

[WaZH01] K. Wang, S. Zhou, and Y. He. **Hierarchical classification of real life documents**. *Proceedings of the 1st SIAM International Conference on Data Mining*. 2001.

[WiFr05] I. Witten and E. Frank. **Data Mining: Practical machine learning tools and techniques, 2nd Edition**. San Francisco: Morgan Kaufmann. 2005.

[WiPW95] E. Wiener, J. Pedersen, and A. Weigend. **A neural network approach to topic spotting**. *Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval*, pp. 317-332. 1995.

[WiRS06] W. Wilbur, A. Rzhetsky, and H. Shatkay. **New directions in biomedical text annotation: definitions, guidelines and corpus construction**. *BMC Bioinformatics*, 7, pp. 356. 2006.

[WiYa96] W. Wilbur and Y. Yang. **An analysis of statistical term strength and its use in the indexing and retrieval of molecular biology texts**. *Computers in Biology and Medicine*, 26(3), pp. 209-222. 1996.

[YaCh92] Y. Yang and C. Chute. **A Linear Least Squares Fit mapping method for information retrieval from natural language texts**. *Proceedings of the 14th International Conference on Computational Linguistics (COLING 92)*, pp. 447-453. 1992.

[YaLi99] Y. Yang and X. Liu. **A re-examination of text categorization methods**. *Proceedings of the 22nd ACM International Conference on Research and Development in Information Retrieval (SIGIR-99)*, pp. 42-49. 1999.

[Yang94] Y. Yang. 1994. **Expert network: effective and efficient learning from human decisions in text categorization and retrieval**. *Proceedings of the 17th ACM International Conference on Research and Development in Information Retrieval (SIGIR-94)*, pp. 13-22. 1994.

[Yang97] Y. Yang. **An evaluation of statistical approaches to text categorization**. *Information Retrieval*, 1(1-2), pp. 69-90.1999.

[YaPe97] Y. Yang and J. Pederson. **A comparative study on feature selection in text categorization**. *Proceedings of the 14th International Conference on Machine Learning (ICML)*, pp. 412--420. 1997.

[YaPH99] J. Yang, R. Parekh, and V. Honavar. **DistAl: An inter-pattern distance-based constructive learning algorithm**. *Intelligent Data Analysis*, 3, pp. 55-73. 1999.

[YKKM03] K. Yamamoto, T. Kudo, A. Konagaya, and Y. Matsumoto. **Protein name tagging for biomedical annotation in text**. *ACL Workshop on Natural Language Processing in Biomedicine*, pp. 65-72. 2003.

[ZJXG05] S. Zhu, X. Ji , W. Xu, and Y. Gong. **Multi-labeled classification using Maximum Entropy method**. *Proceedings of the 28th ACM International Conference on Research and Development in Information Retrieval (SIGIR-05)*, pp. 274-281. 2005.

Appendix A

Maximum Entropy and Maximum Likelihood

In this section, we provide the detailed derivations regarding the theory of Maximum Entropy.

A.1 Maximum Entropy Principle

The principle of Maximum Entropy is to find the least informative (the most uncertain) model that also satisfies any given constraints. A mathematical measure of the uncertainty of a conditional probability distribution $p(c \mid d)$ is provided by the conditional entropy [CoTh91]:

$$H(p) \equiv -\sum_{d \in D} \sum_{c \in C} p(d) p(c \mid d) \log p(c \mid d) \tag{A.1}$$

$$\approx -\sum_{d \in D} \sum_{c \in C} \tilde{p}(d) p(c \mid d) \log p(c \mid d), \tag{A.2}$$

where *D* denotes the document space, and *C* denotes the label space.

The problem of Maximum Entropy is to find a distribution $p^*(c \mid d)$ which has the maximum entropy value $H(p^*)$ among all the distributions $p(c \mid d)$ satisfying the given constraints:

$$\sum_{(d,c) \in Oberved(D,C)} \tilde{p}(d,c) f_i(d,c) = \sum_{d \in D} \tilde{p}(d) \sum_{c \in C} p(c \mid d) f_i(d,c).$$

More formally, let *P* represent the space of all conditional probability distributions $p(c \mid d)$, and P_{Cons} represent a subset of *P* defined by:

$$P_{Cons} \equiv \{ p \in P \mid E_{\tilde{p}}(f_i) = E_p(f_i),\ for\ i \in [1,2,.....n] \}.$$

The task is to find the distribution p^* such that

$$p^* = \arg\max_{p \in P_{Cons}} H(p) . \tag{A.3}$$

A.2 Exponential Form

We call the problem of finding the distribution $p^* = \arg\max_{p \in P_{Cons}} H(p)$ the primal problem. It is actually a constrained optimization problem, maximizing the entropy $H(p) = -\sum_{d \in D} \sum_{c \in C} \tilde{p}(d) p(c \mid d) \log p(c \mid d)$ subject to the following constraint:

(1) $\sum_{(d,c) \in Observed(D,C)} \tilde{p}(c,d) f_i(d,c) = \sum_{d \in D} \sum_{c \in C} \tilde{p}(d) p(c \mid d) f_i(d,c)$ for i $\in$ [1,2….n] .

That is, we restrict the model distribution to have the expected value for each feature $f_i(d,c)$ as derived from the training data. Since $p(c \mid d)$ is a conditional probability distribution, it must satisfy two additional constraints:

(2) $\sum_{c \in C} p(c \mid d) = 1$ for any document *d* ;

(3) $p(c \mid d) \geq 0$ for any document *d* and category *c* ;

The solution to this optimization problem can be found using the method of Lagrange multipliers. We introduce Lagrange multipliers, $\lambda_1, \lambda_2, ..., \lambda_n$, for the constraint, $\sum_{(d,c) \in Observed(D,C)} \tilde{p}(c,d) f_i(d,c) = \sum_{d \in D} \sum_{c \in C} \tilde{p}(d) p(c \mid d) f_i(d,c)$ for i $\in$ [1, 2, … , n], denoted as $\Lambda \equiv (\lambda_1, \lambda_2, ... \lambda_n)$; and a set of γ_d for the constraint, $\sum_{c \in C} p(c \mid d) = 1$ for any document *d*, denoted as $\gamma \equiv (\gamma_d, d \in D)$. We need not introduce an additional multiplier for the third constraint, $p(c \mid d) \geq 0$, since the logarithm function in the original problem,

$\arg\max_{p \in P_{Cons}} (-\sum_{d \in D} \sum_{c \in C} \tilde{p}(d) p(c \mid d) \log p(c \mid d))$, already enforces $p(c \mid d) \geq 0$. We can then define the constrained optimization problem using the Lagrangian:

$$\xi(p, \gamma, \Lambda) \equiv -\sum_{d \in D} \sum_{c \in C} \tilde{p}(d) p(c \mid d) \log p(c \mid d)$$

$$+ \sum_i \lambda_i (\sum_{d \in D} \sum_{c \in C} \tilde{p}(d) p(c \mid d) f_i(d, c) - \sum_{(d,c) \in Observed(D,C)} \tilde{p}(c, d) f_i(d, c))$$

$$+ \sum_{d \in D} \gamma_d (\sum_{c \in C} p(c \mid d) - 1) . \qquad \text{(A.4)}$$

To solve the primal problem, we need to find the unique saddle point of $\xi(p, \gamma, \Lambda)$, which is a maximum with respect to p , and a minimum with respect to (γ, Λ). We will not provide a detailed explanation for why the saddle point is the solution to the initial problem $\arg\max_{p \in P_{Cons}} (-\sum_{d \in D} \sum_{c \in C} \tilde{p}(d) p(c \mid d) \log p(c \mid d))$. Further information is available in books discussing this subject [CoWe03, Apos69].

We first hold γ and Λ fixed, and maximize $\xi(p, \gamma, \Lambda)$ with respect to $p(c \mid d)$. This is done by setting the partial derivative with respect to $p(c \mid d)$ to zero:

$$\frac{\partial \xi}{\partial p(c \mid d)} = -\tilde{p}(d)(1 + \log(p(c \mid d))) + \sum_i \lambda_i \tilde{p}(d) f_i(d, c) + \gamma_d = 0 . \qquad \text{(A.5)}$$

At this optimum, we find that $p(c \mid d)$ has the parametric form:

$$p^*(c \mid d) = \exp\{\sum_i \lambda_i f_i(d, c)\} \exp(\frac{\gamma_d}{\tilde{p}(d)} - 1) . \qquad \text{(A.6)}$$

According to the second constraint, $\sum_c p(c \mid d) = 1$, we can get

$$\exp(\frac{\gamma_d}{\tilde{p}(d)} - 1) = \frac{1}{\sum_{c \in C} \exp\{\sum_i \lambda_i f_i(d, c)\}} . \qquad \text{(A.7)}$$

We rewrite equation (A.6) as

$$p^*(c \mid d) = \frac{\exp\{\sum_i \lambda_i f_i(d,c)\}}{Z(d)}, \tag{A.8}$$

where Z(d) is the normalizing factor, defined as $Z(d) = \sum_{c \in C} \exp\{\sum_i \lambda_i f_i(d,c)\}$.

A.3 Primal Problem to Dual Problem

We have found the optimal parametric form of $p^*(c \mid d)$, which maximizes $\xi(p, \gamma, \Lambda)$ for a fixed (γ, Λ). Next we need to keep $p^*(c \mid d)$ fixed, and minimize $\xi(p^*, \gamma, \Lambda)$ with respect to (γ, Λ). Substituting p^* back into equation (A.4), we get:

$$\begin{aligned}\xi(p^*, \gamma, \Lambda) &= -\sum_{d \in D} \tilde{p}(d) \sum_{c \in C} p(c \mid d) \sum_i \lambda_i f_i(d,c) + \sum_{d \in D} \tilde{p}(d) \sum_{c \in C} p(c \mid d) \log Z(d) \\ &\quad + \sum_i \lambda_i \Big(\sum_{d \in D} \tilde{p}(d) \sum_{c \in C} p(c \mid d) f_i(d,c) - \sum_{(d,c) \in Observed(D,C)} \tilde{p}(d,c) f_i(d,c) \Big) \\ &= \sum_{d \in D} \tilde{p}(d) \sum_{c \in C} p(c \mid d) \log Z(d) - \sum_{(d,c) \in Observed(D,C)} \sum_i \lambda_i \tilde{p}(d,c) f_i(d,c).\end{aligned} \tag{A.9}$$

Since $\sum_{c \in C} p(c \mid d) = \sum_{c \in C} \tilde{p}(c \mid d) = 1$, we can rewrite equation (A.9) as:

$$\begin{aligned}\xi(p^*, \gamma, \Lambda) &= \sum_{d \in D} \tilde{p}(d) \sum_{c \in C} \tilde{p}(c \mid d) \log Z(d) - \sum_{(d,c) \in Observed(D,C)} \sum_i \lambda_i \tilde{p}(d,c) f_i(d,c) \\ &= \sum_{d \in D} \sum_{c \in C} \tilde{p}(d,c) \log Z(d) - \sum_{(d,c) \in Observed(D,C)} \sum_i \lambda_i \tilde{p}(d,c) f_i(d,c).\end{aligned} \tag{A.10}$$

Because $\tilde{p}$ is the empirical distribution observed from the training data, we have $\sum_{d\in D}\sum_{c\in C}\tilde{p}(d,c) = \sum_{(d,c)\in Observed(D,C)}\tilde{p}(d,c)$, and can rewrite equation (A.10) as:

$$\xi(p^*,\gamma,\Lambda) = -(\sum_{(d,c)\in Observed(D,C)}\sum_i \lambda_i \tilde{p}(d,c) f_i(d,c) - \sum_{(d,c)\in Observed(D,C)}\tilde{p}(d,c)\log Z(d))$$

$$= -\sum_{(d,c)\in Observed(D,C)}\tilde{p}(c,d)\log\frac{\exp\{\sum_i \lambda_i f_i(d,c)\}}{Z(d)}$$

$$= -\sum_{(d,c)\in Observed(D,C)}\tilde{p}(c,d)\log p(c\mid d)\,. \qquad \text{(A.11)}$$

The parameter γ does not appear in equation (A.11), thereby we just need to minimize $\xi(p^*,\gamma,\Lambda)$ as a function of the set of parameters $\Lambda \equiv (\lambda_1,\lambda_2,...\lambda_n)$. We define the function $\psi(\Lambda)$ as:

$$\psi(\Lambda) \equiv \xi(p^*,\gamma,\Lambda)\,, \qquad \text{(A.12)}$$

which is called the dual function. So far, the primal problem of finding the distribution $p^* = \arg\max_{p\in P_{Cons}} H(p)$ is transformed to the dual problem of finding the parameter set $\Lambda^* = \arg\min_{\Lambda}\psi(\Lambda)$. In other words, the primal problem of finding the distribution, p^* that maximizes $H(p)$ is equivalent to finding the parameter set $\Lambda \equiv (\lambda_1,\lambda_2,...\lambda_n)$ that minimizes $\psi(\Lambda)$. A thorough discussion of the duality theorem is beyond the scope of this thesis. We refer readers to a book on the subject [Bert99] for further information about the primal and dual problems, and their close relationship under certain assumptions.

A.4 Maximum Likelihood

The dual function $\psi(\Lambda)$ is actually the negative log likelihood of the model distribution, $p(c\,|\,d) = \frac{\exp\{\sum_i \lambda_i f_i(d,c)\}}{Z(d)}$, given the training dataset, $Observed(D,C)$. That is, the likelihood of the model to generate the set of observed (d,c) pairs. Given a dataset $Observed(D,C)$, the log likelihood of the distribution $p(c\,|\,d)$ is defined by:

$$L(p) \equiv \log \prod_{(d,c)\in Observed(D,C)} p(c\,|\,d)^{\tilde{p}(d,c)} .$$

In the case when the distribution $p(c\,|\,d)$ is of the exponential parametric form, $p(c\,|\,d) = \frac{\exp\{\sum_i \lambda_i f_i(d,c)\}}{Z(d)}$, the log likelihood of the model parameters $\Lambda \equiv (\lambda_1, \lambda_2, ... \lambda_n)$ with respect to the data set $Observed(D,C)$ is:

$$\begin{aligned} L(\Lambda) &= \sum_{(d,c)\in Observed(D,C)} \tilde{p}(d,c) \log\Big(\frac{\exp\{\sum_i \lambda_i f_i(d,c)\}}{Z(d)}\Big) \\ &= -\psi(\Lambda) . \end{aligned} \tag{A.13}$$

Hence, the optimal Λ^* that minimizes the dual function $\psi(\Lambda)$ will also maximize the log likelihood $L(\Lambda)$ for the models of the same exponential form, namely,

$$\begin{aligned} \Lambda^* &= \arg\min_{\Lambda} \psi(\Lambda) \\ &= \arg\max_{\Lambda} L(\Lambda) \cdot \end{aligned} \tag{A.14}$$

So far, we have shown that the constrained maximization of entropy is equivalent to the unconstrained maximization of the likelihood of a set of exponential distributions. The model with the maximum entropy is also the one, among all the models of the same parametric form, that fits the training examples best. The problem of Maximum Entropy turns out to be the problem of maximizing the likelihood of a set of exponential models with respect to the training data.

Appendix B

Derivations about the IIS algorithm in Section 5.2

Here we show how inequality (5.16) in Section 5.2 is derived. We have the difference in the log likelihood between the new model $\Lambda+\delta$ and the old model Λ defined by inequality (5.15):

$$L(\Lambda+\delta)-L(\Lambda)=\sum_{(d,c)\in Observed\ (D,C)}(\sum_i \delta_i f_i(d,c)-\log\frac{\sum_{c\in C}\exp\{\sum_i(\lambda_i+\delta_i)f_i(d,c)\}}{\sum_{c\in C}\exp\{\sum_i \lambda_i f_i(d,c)\}})>0\ , \quad (5.15)$$

where δ_i is the change in λ_i at each step.

We make use of the inequality $-\log\partial \geq 1-\partial$ (a convex[1] function always lies above its tangent), to establish a lower bound on the difference in the log likelihood specified in equation (5.15):

$$L(\Lambda+\delta)-L(\Lambda)\geq\sum_{(d,c)\in Observed\ (D,C)}(\sum_i \delta_i f_i(d,c)+1-\frac{\sum_{c\in C}\exp\{\sum_i(\lambda_i+\delta_i)f_i(d,c)\}}{\sum_{c\in C}\exp\{\sum_i \lambda_i f_i(d,c)\}})$$

$$=\sum_{(d,c)\in Observed(D,C)}(\sum_i \delta_i f_i(d,c)+1-\sum_{c\in C}p(c\,|\,d)\exp\{\sum_i \delta_i f_i(d,c)\})\ . \quad (B.1)$$

If we define $f^{\Sigma}(d,c)\equiv\sum_i f_i(d,c)$, we can rewrite inequality (B.1) as:

$$L(\Lambda+\delta)-L(\Lambda)\geq\sum_{(d,c)\in Observed(D,C)}(\sum_i \delta_i f_i(d,c)+1-\sum_{c\in C}p(c\,|\,d)\exp\{f^{\Sigma}(d,c)\sum_i\frac{\delta_i f_i(d,c)}{f^{\Sigma}(d,c)}\})\ .$$

(B.2)

[1] A function $f(x)$ is convex if $-f(x)$ is concave, i.e. on an interval [a,b] if for any two points x_1 and x_2 in [a,b] and any ∂ where $0<\partial<1$, $f(\partial x_1+(1-\partial)x_2)\leq\partial f(x_1)+(1-\partial)f(x_2)$.

Since *exp* is convex, according to Jensen's inequality – namely, for a probability distribution function *p(x)* (*p(x)*≥ 0, and $\sum_x p(x) = 1$), we have

$$\exp\{\sum_x p(x)q(x)\} \leq \sum_x p(x)\exp\{q(x)\}\,.$$

As a consequence, inequality (B.2) can be rewritten as

$$L(\Lambda+\delta) - L(\Lambda) \geq \underbrace{\sum_{(d,c)\in Observed(D,C)} (\sum_i \delta_i f_i(d,c) + 1 - \sum_{c\in C} p(c\,|\,d)\sum_i \frac{f_i(d,c)}{f^{\Sigma}(d,c)} \exp\{\delta_i f^{\Sigma}(d,c)\})}_{A(\delta|\lambda)} \cdot \tag{5.16}$$

Appendix C

An Example of the Maximum Entropy Model

In this section, we use a simple example to illustrate how a Maximum Entropy classifier works in the area of text categorization. As previously mentioned in Section 5.1, we can use a Boolean vector, $c \equiv (c_1,...,c_m)$, to represent the possible label of a document, where m is the total number of categories; and use a Boolean vector, $d \equiv (t_1^d,...,t_k^d)$, to represent a document, where k is the total number of terms. Our goal is to estimate the model distribution $p(c \mid d)$ given a set of training data *D* associated with labels *C*.

A traditional Maximum Entropy based model typically imposes constraints over the following two statistical properties: the prior probability of each category, and the correlation between terms and categories. Accordingly, we define two types of feature function matrix: one captures the occurrence of each category, denoted as f^P; the other captures the co-occurrence between terms and categories, denoted as f^T. For each document-label pair (d,c), the feature function matrix f^P is defined as:

$$f_i^P(d,c) = c_i,\ 1 \le i \le m, \tag{C.1}$$

where m is the number of categories. The feature function matrix f^T is defined as:

$$f_{ij}^T(d,c) = c_i t_j^d,\ 1 \le i \le m, 1 \le j \le k, \tag{C.2}$$

where m is the number of categories and k is the number of terms.

We define the expected number of occurrences of category i observed from the training data as:

$$E_{\tilde{p}}(f_i^P) = \sum_{d,c \in Observed(D,C)} \tilde{p}(d,c) f_i^P(d,c) = \sum_{d,c \in Observed(D,C)} \tilde{p}(d,c) c_i \,,$$

and the expected number of occurrences of category i predicted by the model as:

$$E_p(f_i^P) = \sum_{d \in D} \tilde{p}(d) \sum_{c \in C} p(c \mid d) f_i^P(d,c) = \sum_{d \in D} \tilde{p}(d) \sum_{c \in C} p(c \mid d) c_i \,.$$

To restrict the category distribution predicted by the model to be the same as the empirical category distribution, we define the following constraints:

$$E_{\tilde{p}}(f_i^P) = E_p(f_i^P),\ 1 \le i \le m, \qquad \text{(C.3)}$$

where m is the number of categories.

Similarly, we define the expected number of co-occurrences between category i and term j derived from the training corpus as:

$$E_{\tilde{p}}(f_{ij}^T) = \sum_{d,c \in Observed(D,C)} \tilde{p}(d,c) f_{ij}^T(d,c) = \sum_{d,c \in Observed(D,C)} \tilde{p}(d,c) c_i t_j^d \,,$$

and the expected number of co-occurrences between category i and term j predicted by the model as:

$$E_p(f_{ij}^T) = \sum_{d \in D} \tilde{p}(d) \sum_{c \in C} p(c \mid d) f_{ij}^T(d,c) = \sum_{d \in D} \tilde{p}(d) \sum_{c \in C} p(c \mid d) c_i t_j^d \,.$$

To make the model consistent with the correlation between terms and categories as observed from the training data, we define the following constraints:

$$E_{\tilde{p}}(f_{ij}^T) = E_p(f_{ij}^T),\ 1 \le i \le m,\ 1 \le j \le k, \qquad \text{(C.4)}$$

where m is the number of categories and k is the number of terms.

As shown in equation (A.4) in Appendix A, with the constraints defined above, we can define the Lagrangian as:

$$
\begin{aligned}
\xi(p,\gamma,\Lambda) \equiv & -\sum_{d\in D}\sum_{c\in C}\tilde{p}(d)\,p(c\,|\,d)\log p(c\,|\,d) \\
& + \sum_{i=1}^{m}\lambda_i^P \Big(\sum_{d\in D}\sum_{c\in C}\tilde{p}(d)\,p(c\,|\,d)\,f_i^{\,P}(d,c) - \sum_{(d,c)\in Observed(D,C)}\tilde{p}(c,d)\,f_i^{\,P}(d,c)\Big) \\
& + \sum_{i=1}^{m}\sum_{j=1}^{k}\lambda_{ij}^T \Big(\sum_{d\in D}\sum_{c\in C}\tilde{p}(d)\,p(c\,|\,d)\,f_{ij}^{\,T}(d,c) - \sum_{(d,c)\in Observed(D,C)}\tilde{p}(c,d)\,f_{ij}^{\,T}(d,c)\Big) \\
& + \sum_{d\in D}\gamma_d\Big(\sum_{c\in C}p(c\,|\,d)-1\Big),
\end{aligned}
\tag{C.5}
$$

where λ^P , λ^T , and γ are Lagrange multipliers. λ^P is an $m\times 1$ matrix, where each element λ_i^P represents the weight of the feature $f_i^{\,P}$; λ^T is an $m\times k$ matrix, where each element λ_{ij}^T represents the weight of the feature f_{ij}^T . From equation (A.8) in Appendix A, we know that the optimal probability $p^*(c\,|\,d)$ has the following parametric form:

$$
p^*(c\,|\,d) = \frac{1}{Z(d)}\exp\Big(\sum_{i=1}^{m}\lambda_i^P f_i^{\,P}(d,c) + \sum_{i=1}^{m}\sum_{j=1}^{k}\lambda_{ij}^T f_{ij}^{\,T}(d,c)\Big),
\tag{C.6}
$$

where Z(d) is the normalization factor, defined as :

$$
Z(d) = \sum_{c\in C}\exp\Big(\sum_{i=1}^{m}\lambda_i^P f_i^{\,P}(d,c) + \sum_{i=1}^{m}\sum_{j=1}^{k}\lambda_{ij}^T f_{ij}^{\,T}(d,c)\Big).
$$

From the definition of the features $f^{\,P}$ and $f^{\,T}$, we can also rewrite the Lagrangian as:

$$
\begin{aligned}
\xi(p,\gamma,\Lambda) \equiv & -\sum_{d\in D}\sum_{c\in C}\tilde{p}(d)\,p(c\,|\,d)\log p(c\,|\,d) \\
& + \sum_{i=1}^{m}\lambda_i^P \Big(\sum_{d\in D}\sum_{c\in C}\tilde{p}(d)\,p(c\,|\,d)c_i - \sum_{(d,c)\in Observed(D,C)}\tilde{p}(c,d)c_i\Big) \\
& + \sum_{i=1}^{m}\sum_{j=1}^{k}\lambda_{ij}^T \Big(\sum_{d\in D}\sum_{c\in C}\tilde{p}(d)\,p(c\,|\,d)c_i t_j^d - \sum_{(d,c)\in Observed(D,C)}\tilde{p}(c,d)c_i t_j^d\Big) \\
& + \sum_{d\in D}\gamma_d\Big(\sum_{c\in C}p(c\,|\,d)-1\Big);
\end{aligned}
\tag{C.7}
$$

and rewrite the conditional probability $p^*(c \mid d)$ as:

$$p^*(c \mid d) = \frac{1}{Z(d)} \exp(\sum_{i=1}^{m} \lambda_i^P c_i + \sum_{i=1}^{m} \sum_{j=1}^{k} \lambda_{ij}^T c_i t_j^d), \tag{C.8}$$

where Z(d) is the normalization factor, defined as:

$$Z(d) = \sum_{c \in C} \exp(\sum_{i=1}^{m} \lambda_i^P c_i + \sum_{i=1}^{m} \sum_{j=1}^{k} \lambda_{ij}^T c_i t_j^d) .$$

As introduced in Section 5.2, we can use IIS algorithm to find the optimal parameter sets. We start from an initial parameter set $\Lambda(\lambda^P, \lambda^T)$, and at each step, we find an improvement $\delta(\delta\lambda^P, \delta\lambda^T)$, such that the new model $\Lambda + \delta$ yields a higher log likelihood. If we define $f^\Sigma(d,c)$ as:

$$f^\Sigma(d,c) \equiv \sum_{i=1}^{m} c_i + \sum_{i=1}^{m} \sum_{j=1}^{k} c_i t_j^d$$ |204(()]TJ/T,8E69090 0 12.869

00E8Tj ET1,52 3.7439

Example C.1

Suppose that 2/3 of the documents that contain the word *mice* are relevant documents. We can build a classification model based on the following rule:

> *If a document contains the word mice, the probability that it belongs to the relevant category is 2/3, while the probability it belongs to the irrelevant category is 1/3.*
>
> *Otherwise, the probability distribution for the two categories is uniform, 1/2 each.*

This model is a simple maximum entropy model. It is consistent with the known constraints, and makes no assumptions about what is unknown.

We define the category space as consisting of two categories, *relevant* or *irrelevant* documents, and the term space consisting of k terms, $t_1, \ldots, t_k$. To clearly illustrate the principle of Maximum Entropy, we consider the situation when only one term, say t_1 (*mice*), is used to build the classification model. The training data set is shown in Table C.1.

Table C.1. Training examples for a simple Maximum Entropy model.

Example	Label	t_1 (mice)	t_2	…	t_k
1	relevant	1	…	…	…
2	relevant	1	…	…	…
3	relevant	0	…	…	…
4	irrelevant	1	…	…	…
5	irrelevant	0	…	…	…

We use the Boolean vector $c \equiv (c_{relevant}, c_{irrelevant})$ to denote the class label of an example. For each example, we take into account two feature matrices: f^P and f^T. The feature values can be calculated according to equation (C.1) and (C.2). For instance, the feature values for example *1* are:

$$f^P = (c_{relevant}, c_{irrelevant})' = (1, 0)';$$

$$f^T = (c_{relevant} t_1, c_{irrelevant} t_1)' = (1, 0)'.$$

We then use the examples to train the model. The parameters learned by the IIS algorithm, the statistical properties predicted by the model, and the statistical properties derived from the training examples are shown in Tables C.2 and C.3. Table C.4 shows the class labels predicted by the model. We can see that the model captures the statistical properties of the training data, and it complies with the principle of Maximum Entropy as discussed earlier.

Table C.2. The model parameter λ^P, the expected value for f^P predicted by the model, and the expected value for f^P derived from training data.

λ^P	$E_p(f^P)$	$E_{\tilde{p}}(f^P)$
0.105	0.629	0.6
-0.028	0.371	0.4

Table C.3. The model parameter λ^T, the expected value for f^T predicted by the model, and the expected value for f^T derived from training data.

λ^T	$E_p(f^T)$	$E_{\tilde{p}}(f^T)$
0.330	0.416	0.4
-0.351	0.184	0.2

Table C.4. The predicted class labels.

Examples	Label	Probability p(c|d)
1	relevant	0.693
2	relevant	0.693
3	relevant	0.533
4	relevant	0.693
5	relevant	0.533

To make the conditional probability distribution $p^*(c \mid d)$ exactly match the rule we defined in Example C.1, we do not consider the features pertaining to the prior probability of each category, that is, f^P, and build the classification model solely based on the terms in a document. Consequently, the parametric form of $p^*(c \mid d)$ becomes:

$$p^*(c \mid d) = \frac{1}{Z(d)} \exp(\sum_{i=1}^{m} \sum_{j=1}^{k} \lambda_{ij}^T f_{ij}^T (d,c)) ,$$

where Z(d) is the normalization factor, defined as : $Z(d) = \sum_{c \in C} \exp(\sum_{i=1}^{m} \sum_{j=1}^{k} \lambda_{ij}^T f_{ij}^T (d,c))$.

The parameters learned by the IIS algorithm, the statistical properties predicted by the model, and the statistical properties discovered from the training examples are shown in Table C.5. The class labels predicted by the model are shown in Table C.6. We can see that the model exactly agrees with the rule we expected: if a document contains the word mice, the probability that it belongs to the relevant category is 2/3, while the probability it belongs to the irrelevant category is 1/3; otherwise, the probability distribution for the two

Table C.5. The model parameter λ^T, the expected value for f^T predicted by the modified model, and the expected value for f^T derived from training data.

λ^T	$E_p(f^T)$	$E_{\tilde{p}}(f^T)$
0.287	0.4	0.4
-0.405	0.2	0.2

Table C.6. The predicted class labels by the modified model.

Examples	**Label**	**Probability p(c|d)**
1	relevant	0.667
2	relevant	0.667
3	relevant	0.500
4	relevant	0.667
5	relevant	0.500

Appendix D

The Parametric Form of the Maximum Entropy Model for the Fragment Classification

We show here how to derive the parametric form of the model for the fragment classification. In Section 5.3.3, we have defined five types of feature matrices: the occurrence of categories within each category space, f_F^P, f_E^P, f_N^P; the correlation between terms and categories within each category space, f_F^T, f_E^T, f_N^T; the correlation between categories within each category space, f_F^M, f_E^M, f_N^M; the correlation between the *Fragment Focus* and *Fragment Evidence* category spaces, f^R; and the correlation between the *Fragment Focus* and *Sentence Focus* category spaces f^N.

Equation (A.4) in Appendix A shows that the constrained optimization problem can be solved by introducing Lagrange multipliers. Based on the above features, we need to introduce five types of model parameters that reflect the contributions of the features towards the final classification decision: parameters $\lambda_F^P, \lambda_E^P, \lambda_N^P$ represent the weights of the feature matrices f_F^P, f_E^P, f_N^P respectively; parameters $\lambda_F^T, \lambda_E^T, \lambda_N^T$ represent the weights of the feature matrices f_F^T, f_E^T, f_N^T respectively; parameters $\lambda_F^M, \lambda_E^M, \lambda_N^M$ represent the weights of the feature matrices f_F^M, f_E^M, f_N^M respectively; parameter λ^R represents the weight of the feature matrix f^R; parameter λ^N represents the weight of the feature matrix f^N. Similar to equation (A.4) in Appendix A, with the constraints defined in Section 5.3.3, the Lagrangian can be defined as:

$$\xi(p,\gamma,\Lambda) \equiv -\sum_{d\in D}\sum_{c\in C}\tilde{p}(d)p(c\,|\,d)\log p(c\,|\,d)$$

$$+\sum_{i=1}^{3}\lambda_{Fi}^{P}(\sum_{d\in D}\sum_{c\in C}\tilde{p}(d)p(c\,|\,d)f_{Fi}^{P}-\sum_{(d,c)\in Observed\,(D,C)}\tilde{p}(c,d)f_{Fi}^{P})$$

$$+\sum_{i=1}^{4}\lambda_{Ei}^{P}(\sum_{d\in D}\sum_{c\in C}\tilde{p}(d)p(c\,|\,d)f_{Ei}^{P}-\sum_{(d,c)\in Observed\,(D,C)}\tilde{p}(c,d)f_{Ei}^{P})$$

$$+\sum_{i=1}^{3}\lambda_{Ni}^{P}(\sum_{d\in D}\sum_{c\in C}\tilde{p}(d)p(c\,|\,d)f_{Ni}^{P}-\sum_{(d,c)\in Observed\,(D,C)}\tilde{p}(c,d)f_{Ni}^{P})$$

$$+\sum_{i=1}^{3}\sum_{j=1}^{|t_F|}\lambda_{Fij}^{T}(\sum_{d\in D}\sum_{c\in C}\tilde{p}(d)p(c\,|\,d)f_{Fij}^{T}-\sum_{(d,c)\in Observed\,(D,C)}\tilde{p}(c,d)f_{Fij}^{T})$$

$$+\sum_{i=1}^{4}\sum_{j=1}^{|t_E|}\lambda_{Eij}^{T}(\sum_{d\in D}\sum_{c\in C}\tilde{p}(d)p(c\,|\,d)f_{Eij}^{T}-\sum_{(d,c)\in Observed\,(D,C)}\tilde{p}(c,d)f_{Eij}^{T})$$

$$+\sum_{i=1}^{3}\sum_{j=1}^{|t_N|}\lambda_{Nij}^{T}(\sum_{d\in D}\sum_{c\in C}\tilde{p}(d)p(c\,|\,d)f_{Nij}^{T}-\sum_{(d,c)\in Observed(D,C)}\tilde{p}(c,d)f_{Nij}^{T})$$

$$+\sum_{i=1}^{3}\sum_{j=1}^{3}\lambda_{Fij}^{M}(\sum_{d\in D}\sum_{c\in C}\tilde{p}(d)p(c\,|\,d)f_{Fij}^{M}-\sum_{(d,c)\in Observed(D,C)}\tilde{p}(c,d)f_{Fij}^{M})$$

$$+\sum_{i=1}^{4}\sum_{j=1}^{4}\lambda_{Eij}^{M}(\sum_{d\in D}\sum_{c\in C}\tilde{p}(d)p(c\,|\,d)f_{Eij}^{M}-\sum_{(d,c)\in Observed(D,C)}\tilde{p}(c,d)f_{Eij}^{M})$$

$$+\sum_{i=1}^{3}\sum_{j=1}^{3}\lambda_{Nij}^{M}(\sum_{d\in D}\sum_{c\in C}\tilde{p}(d)p(c\,|\,d)f_{Nij}^{M}-\sum_{(d,c)\in Observed(D,C)}\tilde{p}(c,d)f_{Nij}^{M})$$

$$+\sum_{i=1}^{3}\sum_{j=1}^{4}\lambda_{ij}^{R}(\sum_{d\in D}\sum_{c\in C}\tilde{p}(d)p(c\,|\,d)f_{ij}^{R}-\sum_{(d,c)\in Observed(D,C)}\tilde{p}(c,d)f_{ij}^{R})$$

$$+\sum_{i=1}^{3}\sum_{j=1}^{3}\lambda_{ij}^{N}(\sum_{d\in D}\sum_{c\in C}\tilde{p}(d)p(c\,|\,d)f_{ij}^{N}-\sum_{(d,c)\in Observed\,(D,C)}\tilde{p}(c,d)f_{ij}^{N})$$

$$+\sum_{d\in D}\gamma_d(\sum_{c\in C}p(c\,|\,d)-1),$$

where d indicates a fragment; $c \equiv (c_F, c_E, c_N)$ indicates the category label of a fragment over the three category spaces: the *Focus* of a fragment, the *Evidence* of a fragment, and the context of a fragment, i.e. the *Focus* of the sentence.

According to equation (A.8) in Appendix A, the optimal conditional probability $p^*(c\,|\,d)$ has the following parametric form:

$$p^*(c\,|\,d) = \frac{1}{Z(d)}\exp(\lambda_F^P \cdot f_F^P + \lambda_E^P \cdot f_E^P + \lambda_N^P \cdot f_N^P + \lambda_F^M \cdot f_F^M + \lambda_E^M \cdot f_E^M + \lambda_N^M \cdot f_N^M$$

$$+ \lambda^R \cdot f^R + \lambda^N \cdot f^N + \lambda_F^T \cdot f_F^T + \lambda_E^T \cdot f_E^T + \lambda_N^T \cdot f_N^T),$$

where $\lambda \cdot f$ denotes the sum of the pairwise products of the elements in the two matrices, λ and f, formally: $\lambda \cdot f = \sum_{i,j} \lambda_{ij} f_{ij}$, and Z(d) is the normalization factor, defined as :

$$Z(d) = \sum_{c \in C} \exp(\lambda_F^P \cdot f_F^P + \lambda_E^P \cdot f_E^P + \lambda_N^P \cdot f_N^P + \lambda_F^M \cdot f_F^M + \lambda_E^M \cdot f_E^M + \lambda_N^M \cdot f_N^M$$

$$+ \lambda^R \cdot f^R + \lambda^N \cdot f^N + \lambda_F^T \cdot f_F^T + \lambda_E^T \cdot f_E^T + \lambda_N^T \cdot f_N^T).$$

According to the definition of the features in Section 5.3.3, we can rewrite the conditional probability $p^*(c\,|\,d)$ as:

$${}^{*}(\quad|\quad) = \frac{1}{(\quad)}\exp(\sum_{i=1}^{3} {}^{P}_{Fi} \quad {}_{Fi} + \sum_{i=1}^{4} {}^{P}_{Ei} \quad {}_{Ei} + \sum_{i=1}^{3} {}^{P}_{Ni} \quad {}_{Ni} + \sum_{i=1}^{3}\sum_{j=1}^{3} {}^{M}_{Fij} \quad {}_{Fi} \quad {}_{Fj} + \sum_{i=1}^{4}\sum_{j=1}^{4} {}^{M}_{Eij} \quad {}_{Ei} \quad {}_{Ej}$$

$$Z(d)=\sum_{c\in C}\exp(\sum_{i=1}^{3}\lambda_{Fi}^{P}c_{Fi}+\sum_{i=1}^{4}\lambda_{Ei}^{P}c_{Ei}+\sum_{i=1}^{3}\lambda_{Ni}^{P}c_{Ni}+\sum_{i=1}^{3}\sum_{j=1}^{3}\lambda_{Fij}^{M}c_{Fi}c_{Fj}+\sum_{i=1}^{4}\sum_{j=1}^{4}\lambda_{Eij}^{M}c_{Ei}c_{Ej}$$

$$+\sum_{i=1}^{3}\sum_{j=1}^{3}\lambda_{Nij}^{M}c_{Ni}c_{Nj}+\sum_{i=1}^{3}\sum_{j=1}^{4}\lambda_{ij}^{R}c_{Fi}c_{Ej}+\sum_{i=1}^{3}\sum_{j=1}^{3}\lambda_{ij}^{N}c_{Fi}c_{Nj}$$

$$+\sum_{i=1}^{3}\sum_{j=1}^{|t_F|}\lambda_{Fij}^{T}c_{Fi}t_{Fj}^{d}+\sum_{i=1}^{4}\sum_{j=1}^{|t_E|}\lambda_{Eij}^{T}c_{Ei}t_{Ej}^{d}+\sum_{i=1}^{3}\sum_{j=1}^{|t_N|}\lambda_{Nij}^{T}c_{Ni}t_{Nj}^{d}).$$

We can use the IIS algorithm to find the optimal parameter sets. We start from an initial parameter set $\Lambda\equiv(\lambda_F^P,\lambda_E^P,\lambda_N^P,\lambda_F^M,\lambda_E^M,\lambda_N^M,\lambda^R,\lambda^N,\lambda_F^T,\lambda_E^T,\lambda_N^T)$, and at each step, we find an improvement $\delta\equiv(\delta\lambda_F^P,\delta\lambda_E^P,\delta\lambda_N^P,\delta\lambda_F^M,\delta\lambda_E^M,\delta\lambda_N^M,\delta\lambda^R,\delta\lambda^N,\delta\lambda_F^T,\delta\lambda_E^T,\delta\lambda_N^T)$, such that the new model $\Lambda+\delta$ yields a higher log likelihood with respect to the training data. If we define $f^{\Sigma}(d,c)$ as:

$$f^{\Sigma}(d,c)\equiv\sum_{i=1}^{3}c_{Fi}+\sum_{i=1}^{4}c_{Ei}+\sum_{i=1}^{3}c_{Ni}+\sum_{i=1}^{3}\sum_{j=1}^{3}c_{Fi}c_{Fj}+\sum_{i=1}^{4}\sum_{j=1}^{4}c_{Ei}c_{Ej}+\sum_{i=1}^{3}\sum_{j=1}^{3}c_{Ni}c_{Nj}$$

$$+\sum_{i=1}^{3}\sum_{j=1}^{4}c_{Fi}c_{Ej}+\sum_{i=1}^{3}\sum_{j=1}^{3}c_{Fi}c_{Nj}+\sum_{i=1}^{3}\sum_{j=1}^{|t_F|}c_{Fi}t_{Fj}^{d}+\sum_{i=1}^{4}\sum_{j=1}^{|t_E|}c_{Ei}t_{Ej}^{d}+\sum_{i=1}^{3}\sum_{j=1}^{|t_N|}c_{Ni}t_{Nj}^{d},$$

according to equation (5.17) in Section 5.2, the best δ at each step should satisfy:

$$\sum_{(d,c)\in Observed(D,C)}\Big(c_{Fi}-\sum_{c\in C}p(c\,|\,d)c_{Fi}\exp\{\delta\lambda_{Fi}^{P}f^{\Sigma}(d,c)\}\Big)=0,\ 1\le i\le 3;$$

$$\sum_{(d,c)\in Observed(D,C)}\Big(c_{Ei}-\sum_{c\in C}p(c\,|\,d)c_{Ei}\exp\{\delta\lambda_{Ei}^{P}f^{\Sigma}(d,c)\}\Big)=0,\ 1\le i\le 4;$$

$$\sum_{(d,c)\in Observed(D,C)}\Big(c_{Ni}-\sum_{c\in C}p(c\,|\,d)c_{Ni}\exp\{\delta\lambda_{Ni}^{P}f^{\Sigma}(d,c)\}\Big)=0,\ 1\le i\le 3;$$

$$\sum_{(d,c)\in Observed(D,C)}\Big(c_{Fi}c_{Fj}-\sum_{c\in C}p(c\,|\,d)c_{Fi}c_{Fj}\exp\{\delta\lambda_{Fij}^{M}f^{\Sigma}(d,c)\}\Big)=0,\ 1\le i<j\le 3;$$

$$\sum_{(d,c)\in Observed(D,C)}\Big(c_{Ei}c_{Ej}-\sum_{c\in C}p(c\,|\,d)c_{Ei}c_{Ej}\exp\{\delta\lambda_{Eij}^{M}f^{\Sigma}(d,c)\}\Big)=0,\ 1\le i<j\le 4;$$

$$\sum_{(d,c)\in Observed(D,C)} \left(c_{Ni} c_{Nj} - \sum_{c\in C} p(c \mid d) c_{Ni} c_{Nj} \exp\{\delta\lambda_{Nij}^{M} f^{\Sigma}(d,c)\} \right) = 0\,,\ 1 \le i < j \le 3\,;$$

$$\sum_{(d,c)\in Observed(D,C)} \left(c_{Fi} c_{Ej} - \sum_{c\in C} p(c \mid d) c_{Fi} c_{Ej} \exp\{\delta\lambda_{ij}^{R} f^{\Sigma}(d,c)\} \right) = 0\,,\ 1 \le i \le 3,\ 1 \le j \le 4\,;$$

$$\sum_{(d,c)\in Observed(D,C)} \left(c_{Fi} c_{Nj} - \sum_{c\in C} p(c \mid d) c_{Fi} c_{Nj} \exp\{\delta\lambda_{ij}^{N} f^{\Sigma}(d,c)\} \right) = 0\,,\ 1 \le i \le 3,\ 1 \le j \le 3\,;$$

$$\sum_{(d,c)\in Observed(D,C)} \left(c_{Fi} t_{Fj}^{d} - \sum_{c\in C} p(c \mid d) c_{Fi} t_{Fj}^{d} \exp\{\delta\lambda_{Fij}^{T} f^{\Sigma}(d,c)\} \right) = 0\,,\ 1 \le i \le 3,\ 1 \le j \le \left| t_F \right|\,;$$

$$\sum_{(d,c)\in Observed(D,C)} \left(c_{Ei} t_{Ej}^{d} - \sum_{c\in C} p(c \mid d) c_{Ei} t_{Ej}^{d} \exp\{\delta\lambda_{Eij}^{T} f^{\Sigma}(d,c)\} \right) = 0\,,\ 1 \le i \le 4, 1 \le j \le \left| t_E \right|\,;$$

$$\sum_{(d,c)\in Observed(D,C)} \left(c_{Ni} t_{Nj}^{d} - \sum_{c\in C} p(c \mid d) c_{Ni} t_{Nj}^{d} \exp\{\delta\lambda_{Nij}^{T} f^{\Sigma}(d,c)\} \right) = 0\,,\ 1 \le i \le 3, 1 \le j \le \left| t_N \right|.$$

We can use root finding procedures to solve the above equations. The process repeats until the log likelihood of the model converges.

Appendix E

Stop Word List

Table E.1. Stop Word List. F denotes Focus, P denotes Polarity, C denotes Certainty, E denotes Evidence, and T denotes Trend. 1 indicates a word is defined as a stop word, and 0 indicates otherwise.

Word	F	P	C	E	T	Word	F	P	C	E	T	Word	F	P	C	E	T	Word	F	P	C	E	T
a	1	1	1	1	1	enough	1	1	0	0	1	mostly	1	0	0	0	1	their	1	1	1	0	1
about	1	1	0	1	1	et	0	1	0	0	1	mr	1	1	1	1	1	them	1	1	1	0	1
above	1	1	1	1	1	etc	1	1	1	1	1	much	1	0	0	0	1	themselves	1	1	0	1	1
across	1	1	1	1	1	ever	1	0	0	1	1	must	1	0	0	0	1	then	1	1	1	1	1
after	1	1	1	1	1	every	1	0	0	1	1	my	1	1	1	1	1	thence	1	1	1	1	1
afterwards	1	1	1	0	1	everyone	1	1	0	1	1	myself	1	1	1	1	1	there	0	1	1	1	1
again	1	1	0	0	1	everything	1	1	0	1	1	namely	1	1	1	1	1	thereafter	1	1	1	1	1
against	1	0	1	1	0	everywhere	1	1	0	1	1	neither	1	0	0	0	1	thereby	1	1	0	1	1
al	0	1	0	0	1	except	1	0	0	1	1	never	1	0	0	0	1	therefore	1	1	0	0	1
all	1	1	1	1	1	find	1	1	0	0	1	nevertheless	0	1	1	1	1	therein	1	1	1	1	1
almost	1	1	0	0	1	for	1	1	1	1	1	next	1	1	1	1	1	thereupon	1	1	0	0	1
alone	1	0	1	1	1	found	1	1	0	0	1	no	1	0	0	0	1	these	1	1	1	1	1
along	1	1	1	1	1	from	1	1	1	1	1	nobody	1	0	0	0	1	they	1	1	1	0	1
already	1	1	0	0	1	further	1	1	1	1	1	noone	1	0	0	0	1	this	1	1	1	1	1
also	1	1	1	1	1	get	1	1	1	1	0	nor	1	0	0	0	1	thorough	1	1	0	0	1
although	1	1	0	0	1	give	1	1	1	1	0	not	1	0	0	0	1	those	1	1	1	1	1
always	1	0	0	0	1	go	1	1	1	1	0	nothing	1	0	0	0	1	though	1	1	1	1	1
am	1	1	1	1	1	gov	1	1	1	1	1	now	1	1	1	1	1	through	1	1	1	1	1
among	1	1	1	1	1	had	1	0	1	1	0	nowhere	1	0	1	1	1	throughout	1	1	1	1	1
amongst	1	1	1	1	1	has	1	0	1	1	0	of	1	1	1	1	1	thru	1	1	1	1	1
an	1	1	1	1	1	have	1	0	1	1	0	off	1	1	0	1	1	thus	1	1	0	0	1
analyze	1	1	1	1	1	he	1	1	1	1	1	often	1	1	1	1	1	to	1	1	1	1	1
and	1	1	1	1	1	hence	1	1	0	0	1	on	1	1	1	1	0	together	1	1	1	1	1
another	1	1	1	1	1	her	1	1	1	1	1	only	1	0	1	1	1	too	1	0	1	1	1
any	1	0	0	0	1	here	1	1	1	1	1	onto	1	1	1	1	1	toward	1	1	1	1	0
anyhow	1	1	1	1	1	hereafter	1	1	0	0	1	or	1	1	1	1	1	towards	1	1	1	1	0
anyone	1	0	0	0	1	hereby	1	1	0	0	1	other	1	1	1	1	1	try	1	1	1	1	1
anything	1	0	0	0	1	herein	1	1	0	0	1	others	1	1	1	1	1	type	1	1	1	1	1
anywhere	1	0	0	0	1	hereupon	1	1	0	0	1	otherwise	1	1	1	1	1	ug	1	1	1	1	1
applicable	1	0	1	1	0	hers	1	1	1	1	1	our	1	1	1	1	1	under	1	1	1	1	0
apply	1	1	1	1	1	herself	1	1	1	1	1	ours	1	1	1	1	1	unless	1	1	1	1	0
are	1	0	1	1	1	him	1	1	1	1	1	ourselves	1	1	1	1	1	until	1	1	1	1	0
around	1	1	0	0	1	himself	1	1	1	1	1	out	1	1	1	1	1	up	1	1	1	1	0
as	1	1	0	0	1	his	1	1	1	1	1	over	1	1	1	1	1	upon	1	1	1	1	0
assume	1	1	0	0	1	how	1	1	1	1	1	own	1	1	1	1	1	us	1	1	1	0	1
at	1	1	1	1	1	however	1	0	1	1	1	oz	1	1	1	1	1	used	0	1	1	0	1

Word	F	P	C	E	T	Word	F	P	C	E	T	Word	F	P	C	E	T	Word	F	P	C	E	T
be	1	0	0	0	0	hr	1	1	1	1	1	per	1	1	1	1	1	using	0	1	1	0	1
became	1	1	1	1	0	ie	1	1	1	1	1	perhaps	1	1	0	0	1	various	1	1	1	1	1
because	1	1	0	0		if	1	1	1	1	1	pm	1	1	1	1	1	very	1	1	0	0	1
become	1	1	1	1	0	ii	1	1	1	1	1	precede	1	1	1	1	1	via	1	1	1	1	1
becomes	1	1	1	1	0	iii	1	1	1	1	1	presently	1	1	1	1	1	was	1	0	1	1	1
becoming	1	1	1	1	0	in	1	1	1	1	1	previously	1	1	1	1	1	we	1	1	0	0	1
been	1	0	1	1	1	inc	1	1	1	1	1	pt	1	1	1	1	1	were	1	0	1	1	1
before	1	1	1	1	1	incl	1	1	1	1	1	rather	1	0	1	1	1	what	1	1	1	1	1
beforehand	1	1	1	1	1	indeed	1	1	0	1	1	regarding	1	1	1	1	1	whatever	1	0	0	0	1
being	1	0	1	1	0	into	1	1	1	1	1	relate	1	1	1	1	1	when	1	1	1	1	1
below	1	1	1	1	1	investigate	1	1	0	0	1	said	1	1	1	1	1	whence	1	1	1	1	1
beside	1	1	1	1	1	is	1	0	0	0	0	same	1	1	1	1	1	whenever	1	1	1	1	1
besides	1	1	1	1	1	it	1	1	1	1	1	seem	1	1	0	0	1	where	1	1	1	1	1
between	1	1	1	1	1	its	1	1	1	1	1	seemed	1	1	0	0	1	whereafter	1	1	1	1	1
beyond	1	0	1	1	1	itself	1	1	1	1	1	seeming	1	1	0	0	1	whereas	1	1	1	1	1
both	1	1	1	1	1	j	1	1	1	1	1	seems	1	1	0	0	1	whereby	1	1	1	1	1
but	1	1	1	1	1	jour	1	1	1	1	1	seriously	1	1	1	1	1	wherein	1	1	1	1	1
by	1	1	1	1	1	journal	1	1	1	1	1	several	1	1	1	1	1	whereupon	1	1	0	0	1
came	1	1	1	1	1	just	1	0	0	1	1	she	1	1	1	1	1	wherever	1	1	1	1	1
cannot	1	0	0	0	0	kg	1	1	1	1	1	should	1	1	1	1	1	whether	1	1	1	1	1
cc	1	1	1	1	1	last	1	1	1	1	1	show	0	0	0	0	0	which	1	1	1	1	1
cm	1	1	1	1	1	latter	1	1	1	1	1	showed	0	0	0	0	0	while	1	1	1	1	1
come	1	1	1	1	1	latterly	1	1	1	1	1	shown	0	0	0	0	0	whither	1	1	1	1	1
compare	1	1	1	1	1	lb	1	1	1	1	1	since	1	1	1	1	1	who	1	1	0	0	1
could	1	0	0	0	0	ld	1	1	1	1	1	so	1	1	0	0	1	whoever	1	1	0	1	1
de	1	1	1	1	1	letter	1	1	1	1	1	some	1	1	1	1	1	whom	1	1	0	1	1
dealing	1	1	1	1	1	like	1	1	0	0	1	somehow	1	1	0	0	1	whose	1	1	1	1	1
department	1	1	1	1	1	ltd	1	1	1	1	0	someone	1	1	1	1	1	why	1	1	1	1	1
depend	1	1	1	1	0	made	1	1	1	1	0	something	1	1	1	1	1	will	1	1	1	1	1
did	1	0	1	1	0	make	1	1	1	1	0	sometime	1	1	0	0	1	with	1	1	1	1	1
discover	0	1	0	0	1	many	1	1	1	1	1	sometimes	1	1	0	0	1	within	1	1	1	1	1
dl	1	1	1	1	1	may	1	0	0	0	1	somewhere	1	1	0	0	1	without	1	0	1	1	1
do	1	0	1	1	1	me	1	1	1	1	1	still	1	1	1	1	1	wk	1	1	1	1	1
does	1	0	1	1	1	meanwhile	1	1	1	1	1	studied	0	1	1	0	1	would	1	0	1	1	1
during	1	1	1	1	1	mg	1	1	1	1	1	sub	1	1	1	1	1	wt	1	1	1	1	1
each	1	0	0	1	1	might	1	0	0	0	1	such	1	1	1	1	1	yet	1	1	1	1	1
ec	1	1	1	1	1	ml	1	1	1	1	1	take	1	1	1	1	1	you	1	1	1	1	1
ed	1	1	1	1	1	mm	1	1	1	1	1	tell	1	1	1	1	1	your	1	1	1	1	1
effected	1	1	1	1	0	mo	1	1	1	1	1	th	1	1	1	1	1	yours	1	1	1	1	1
eg	1	1	1	1	1	more	1	1	1	1	1	than	1	1	1	1	1	yourself	1	1	1	1	1
either	1	0	1	1	1	moreover	1	1	0	0	1	that	1	1	1	1	1	yourselves	1	1	1	1	1
else	1	1	1	1	1	most	1	0	0	0	1	the	1	1	1	1	1	yr	1	1	1	1	1
elsewhere	1	1	1	1	1																		