ONTOLOGY BASED TEXT CLASSIFICATION

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF COMPUTER SCIENCE AND ENGINERING IN THE FACULTY OF ENGINEERING AND TECHNOLOGY, JADAVPUR UNIVERSITY 2016

By

Abhranil Chatterjee

Registration No.: 128999 of 2014-15

Examination Roll No.: M4CSE1613

Under the Guidance of

Dr. Diganta Saha

Department of Computer Science and Engineering

Jadavpur University, Kolkata-700032

India

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

TO WHOM IT MAY CONCERN

I hereby recommend that the thesis entitled **"ONTOLOGY BASED TEXT CLASSIFICATION"** prepared under my supervision by **Abhranil Chatterjee** (Registration No. 128999 of 2014-15 , Examination Roll No. M4CSE1613), may be accepted in partial fulfillment for the degree of Master of Computer Science and Engineering in the Faculty of Engineering and Technology,Jadavpur University.

……

Dr. Diganta Saha(Thesis Supervisor)

Department of Computer Science and Engineering

Jadavpur University, Kolkata-32

Countersigned

…….

Prof. Debesh Kumar Das

Head, Department of Computer Science and Engineering,

Jadavpur University, Kolkata-32.

…….

Prof. Sivaji Bandyopadhyay

Dean, Faculty of Engineering and Technology,

Jadavpur University, Kolkata-32.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

CERTIFICATE OF APPROVAL

The foregoing thesis is hereby accepted as a credible study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the thesis only for the purpose for which it is submitted.

…….

Signature of Examiner 1

Date:

…….

Signature of Examiner 2

Date:

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

Declaration of Originality and Compliance of Academic Ethics

I hereby declare that this thesis entitled "ONTOLOGY BASED TEXT CLASSIFICATION" contains literature survey and original research work by the undersigned candidate, as part of his Degree of Master of Computer Science & Engineering. All information have been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name: Abhranil Chatterjee

Registration No: 12899 of 2014-15

Exam Roll No.: M4CSE1613

Thesis Title: Ontology Based Text Classification

……

Signature with Date

Acknowledgement

I express my honest and sincere thanks and humble gratitude to my respected teacher and guide Dr. Diganta Saha, for his exclusive guidance and entire support in completing and producing this project successfully. I am very much indebted to him for the constant encouragement and continuous inspiration that he has given to me. I am deeply grateful to her for the long discussions that helped to enrich the technical content of this manuscript. Without her enthusiasm, encouragement, support and continuous optimism this thesis would hardly have been completed. The above words are only a token of my deep respect towards him for all he has done to take my project to the present shape.

I would like to thank Prof. Debesh Kumar Das, Head, Department of Computer Science and Engineering, Jadavpur University also Prof. Sivaji Bandyopadhyay, Dean, Faculty of Engineering and Technology, Jadavpur University for providing me with moral support at times of need.

This thesis would not have been completed without the inspiration and support of a number of wonderful individuals — my thanks and appreciation to all of them for being part of this journey and making this thesis possible.

Finally I convey my real sense of gratitude and thankfulness to all my friends and family members for their unconditional support without which I would hardly be capable of producing this huge work.

……

(Signature)

Abhranil Chatterjee

Registration No: 128999 of 2014-15

Exam Roll No.: M4CSE1613

Department of Computer Science & Engineering

Jadavpur University

Contents

Chapter 1

Introduction

Text documents are the most dominant and useful information resource in today's world. Moreover, the rapid growth of digital content makes it essential to categorize the documents for efficient retrieval of relevant information. In the field of information retrieval and text mining researchers seek to find categories of text documents. The two main approaches of finding the categories are to group similar documents into a meaningful set also called unsupervised categorization or text clustering or to assign a document to a class from the predefined list of categories also called supervised categorization or text classification'. Formally text classification is the machine learning task of assigning a document to one of the pre-defined document category or class based on the content of the document. In classification process a large set of documents are used for training and designing a classifier which is used to label the text documents. The various applications of text classification can be found in the areas of electronic news classification, webpage classification spam filtering and many more. Moreover, the rapid growth of digital content makes it essential to categorize the documents for efficient retrieval of relevant information.

However, both text clustering and text classification process use the 'bag of words' model where "a text (such as a sentence or a document) is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity". It has been seen that the vocabulary ambiguity of natural language is major drawback of this traditional bag of words model which reduces the accuracy of the text classification process. For an example we can think of the ambiguities caused by homonyms and

synonyms. Homonyms are the words that have same spelling and same pronunciation but have different meaning. Say, the word 'bank' that can have different meaning on different discipline, that can refer to the edge of a river also can refer to financial institution. Hence, any document containing the word 'bank' can be misclassified. On the other hand, synonyms are the words that have different appearance but same meaning, for example 'cosmology' and 'astronomy'. We can use 'ontology' to make the system aware of these ambiguities and improve the classification rate [1, 3, 4].

In this paper after discussing the previous works that has been made in this field and our proposed system,we start our discussion with traditional bag of words model in details. Then we divert our attention to the discussion of ontology where we give the formal mathematical definition of ontology and then an example is shown to clarify the use of ontology. In next section we discuss how ontology can be used with bag of words model which can improve the overall accuracy. We used 'Lucene' with 'Wordnet3.1' to implement this idea and then we compare the experimental results using and without using ontology and show the difference.

Chapter 2

Previous Work

Both linguistics and natural language processing have addressed the issue of the existence of the ambiguities of natural languages. It is reflected in the work of Richardson and Smeaton, 1995. In 1985, Princeton University started the WordNet project (Miller et al., 1990, Morato et al, 2004) to develop a lexical resource for the English language. Many other electronic lexical resources have been developed (Best, Nathan, and Lebiere, 2010; Prevot, Borgo, and Oltramari, 2005; Valitutti, Strapparava and Stock, 2004). Various attempts to use lexical resources (or ontologies) for automatic classification are also evident. Prabowo et al. (2002) and Song et al. (2005) report ontology based systems for classification of Web pages. The former used ontologies based on the Library of Congress Classification (LCC) and the Dewey Decimal Classification (DDC) schemes. However, according to Song et al. (2005), Prabowo et al.s (2002) work is less productive as it is not adaptive to creating a sophisticated classification. Song et al. (2005) developed their ontology semi-automatically in the domain of economy. However, as this ontology was not so descriptive, one cannot expect highly accurate results for classification. In addition to the Web page classification, ontologies are also applied in classification of emails and news in digital format. Taghva et al. (2003) formulated an ontology based classification system for emails. Tenenboim, Shapira, and Shoval (2008) report an ontology based classification system for electronic newspapers. Bloehdorn et al. of University of Karlsruhe described an integrated framework OTTO(Ontology-based Textmining Framework).

Chapter 3

Proposed System

In this paper we want to show how we can improve the accuracy of the text classification process using Ontology. For indexing purpose, instead of representing each document with a vector of large number of terms(dimension) where each document consists of only a portion of the terms selected as feature, we can use inverted index where each term consists of a list of documents where it occurs. Moreover, to implement the ontological structure Wordnet will be used. Though the use of Wordnet as Ontology can be questioned, but it should be noted that an ontology consists of(that is discussed in detail later in this paper) the taxonomy(i.e the hierarchy of the concepts) and the relation between the concepts. For classification purpose, we are only interested in the hierarchy of concepts of that Ontology and Wordnet can be thought of as the hierarchy of concepts. In this paper we are not interested with the performance of the classifier. Hence, the result will not be compared using various classifiers. Instead of other classifiers, only K-Nearest Neighbor, the simplest classifier will be used to classify the documents and to show the improvement in accuracy using ontology based text classification. In this paper, we keep our main focus on the use of ontogical information in text classification and how to improve it over conventional classification technique.

Chapter 4

Indexing Using Vector Space Model

To classify the documents we need to search and retrieve those documents, however we do not need to work with the documents directly. Instead a different strategy is followed to represent the semantic aspects of the documents which is known as indexing. In this process the contents of the documents are represented as a set of indexing features. As we are dealing only with the text documents, in our case the indexing units are the words.

4.1 Indexing Process

Nowadays from text categorization to information retrieval, all the systems rely on an automatic indexing of documents. A simple automatic indexing algorithm is composed of four steps:[2]
1.Structure analysis and tokenization
2.Stopword removal
3.Morphological normalization
4.Weighting

4.1.1 Structure Analysis and Tokenization

Structure analysis is the process of parsing the documents to recognize their structures such as title, section or paragraph. For each of these structures the documents are segmented into words or tokens which is called tokeniza-

tion. Here decisions must be made regarding numbers, special characters, hyphenation, and capitalization. We also need to take care of various short forms and multiword expressions.

4.1.2 Stopword Removal

In this step very frequent word forms such as determiners the, prepositions to, conjunctions and, pronouns you and some verbal forms is etc. appearing in a stopword list are usually removed. This step is important as stopwords do not bear much meaning, represent noise and reduce the accuracy of the classification performance, since they do not discriminate between relevant and non relevant documents. Moreover removing the stopwords we can reduce the number of indexing features significantly and that helps to speed up the classification process.Although the objectives seem clear, there is no clear and complete methodology to develop a stopword list. Furthermore, some expressions, as The Who, and-or gates, or vitamin A, based on words usually found in stopword list, are very useful in specifying more precisely what the user wants. Similarly, after converting all characters into lowercase letters, some ambiguity can be introduced as, for example, with the expressions US citizen viewed as us citizen or IT scientist as it scientist, as both us and it are usually considered stopwords. The strategy regarding the treatment of stopwords may thus be refined by identifying that US and IT are not pronouns in the above examples, e.g., through a part-of-speech tagging step.

4.1.3 Morphological Normalization

As a third step, an indexing procedure uses some type of morphological normalization in an attempt to detect the root word and conflate word variants into the same root or stem. Stemming procedures, which aim to identify the stem of a word and use it in lieu of the word itself, are by far the most common morphological normalization procedures used in indexing. Grouping words having the same root under the same stem may increase the success rate of classification. Assuming that words with the same stem refer to the same idea or concept and must be therefore indexed under the same form, increases the effectiveness of the classification. For example the terms 'play' , 'playing' , 'played' can be morphologically normalized to the same stem 'play'.

4.1.4 Weighting

As we already discusses, an indexing process segments a document into words removing all the stopwords that play a little role in categorizing and stripping the suffixes to produce a set of indexing units. This result corresponds to a binary indexing scheme within which each document is represented by a set of stemmed keywords without any weight assigned. Of course we may consider additional indexing rules as, for example, to consider only the main aspects of each document. To achieve this, we can consider as indexing units only terms appearing more often than a given threshold. Binary logical restrictions may often be too restrictive for a document and query indexing. It is not always clear whether or not a document should be indexed by a given term. Often, a more appropriate answer is neither yes nor no, but rather something in between. Term weighting creates a distinction among terms and increases indexing flexibility. Thus we need to assign higher weight to more important features and lower weight to marginal ones. To weight appropriately each indexing unit, we may consider three components, namely, the term frequency, the inverse document frequency.

First, one can assume that an indexing unit appearing more often in a document must have a higher importance in describing its semantic content. We can measure this influence by counting its term frequency i.e., its number of occurrences within a document, a value denoted tf(t , d) that is the frequency of the term t in document d. Thus, if a term occurs three times in a document, its tf will be 3. Of course, one can consider other simple variants, especially when considering that the occurrence of a given term in a document is a rare event. Thus, it may be good practice to give more importance to the first occurrence than to the others. To do so, the tf component is sometimes computed as $log(tf + 1)$ or as $0.5 + 0.5[tf/max(tf)]$. In this latter case, the normalization procedure is obtained by dividing tf by the maximum tf value for any term in that document.

As a second weighting component, one may consider that those terms occurring very frequently in the collection do not help us discriminate between relevant and nonrelevant documents. For example, the query computer database is likely to yield a very large number of articles from a collection about computer science. We meet here the notion of term frequency in the collection (i.e., the number of documents in which a term appears), a value denoted df

, and called document frequency. More precisely, we will use the logarithm of the inverse document frequency (denoted by $idf = log(n/df)$, with n indicating the number of documents in the collection), resulting in more weight for rare words and less weight for more frequent ones. With this component, if a term occurs in every document ($df = n$), its weight will be $log(n/n) = 0$, and thus will be ignored. On the other hand, when a term appears in only one document ($df = 1$), its weight will reach the maximum for the collection, namely, $log(n/1) = log(n)$.To integrate both components (tf and idf), we can multiply the weight corresponding to the importance of the indexing term within the document (tf) by its importance considering the whole collection (idf). We thus obtain the well-known $tfidf$ formula.

4.2 Vector Space Model

The abovementioned procedures are used to index a document and each document is represents as a set of weighted indexing terms that improves the efficiency to access information. In Vector Space model each document is represented as a multidimensional vector where the indexing terms are treated as one dimension. For example, let us assume we have a set of documents D with cardinality(the number of documents) M and after indexing we get a set of indexing terms T with cardinality(the number of indexing terms for the total set D) N. Then each document of D is represented as a N-dimensional vector. In that vector representation, we can only keep the binary values that denotes the existence of that particular term in that particular document or we can also keep fractional weights indicating the 'relative importance' of that term in that document. The relative importance of a term t in a document d is measured by tf-idf(t , d) as we already mentioned. The set of indexing terms forms linearly independent basis vectors. We assume therefore that the indexing terms are independent of one another or in other words, presence of a term does not depend on other terms. For example, while indexing 'Jadavpur University', both the terms 'Jadavpur' and 'University' are treated as independent. Though this representaion is a simplified assumption, it performs well.

Based on a geometric intuition, the vector-space model does not have a solid and precise theory that is able to clearly justify some of its aspects like computing the degree of similarity or the distance between the documents. We

adopt different formulas to measure the similarity. If we denote by w_{ij} the weight of the indexing term t_j in the document D_i , and by w_{kj} the weight of the same term in the document D_k. The similarly between these documents could be computed as follows:

$$sim(D_i, D_k) = \sum_{j=1}^{N} w_{ij} w_{kj}$$

Of course, the vector representing D_i and D_k is composed of N values with N representing the number of distinct indexing terms. As an alternative similarity measure, we may compute the cosine of the angle between the vectors representing D_i and D_k as follows:

$$sim(D_i, D_k) = \frac{\sum_{j=1}^{N} w_{ij} w_{kj}}{\sqrt{\sum_{j=1}^{N} w_{ij}^2} \sqrt{\sum_{j=1}^{N} w_{kj}^2}}$$

In order to avoid computing all elements expressed in the previous formula at retrieval time, we may store the weights associated with each element of D_i in the inverted file. If we apply the well-known weighting scheme tf idf , we can compute and store the weight w_{ij} of each indexing term t_j for the document D_i during the indexing as follows:

$$sim(D_i, D_k) = \frac{tf_{ij} idf_j}{\sqrt{\sum_{j=1}^{N} (tf_{ij} idf_j)^2}}$$

Advanced weighting formulas have been proposed within the vector-space model leading to different formulas (Buckley et al., 1996), some being more effective than others. Moreover, various attempts have been suggested to account for term dependencies (Wong et al., 1987). Most of these attempts can be seen as transformations aiming at expanding document representation through a linear transformation T: the vector D_i becomes TD_i . Often, the matrix T represents a termterm similarity matrix, which can be defined by compiling some a priori given thesaurus, or by automatically building a semantic similarity matrix. In particular, the Generalized Vector-Space Model (GVSM) (Wong et al., 1987) corresponds to setting T to the term-document matrix (i.e., the transpose of the document-term matrix).

Chapter 5

Classification

After the indexing process each document is represented as a N dimensional vector where N is the number of indexing terms. To use a classifier, a set of documents are used for training purpose where the category or the class of the document is assigned manually. Then the classifier is able to categorize the document automatically to the class it belongs.There are various classifier that we can use like 'k-nearest neighbor' , 'naive Bayes classifier' , 'support vector machine' etc.. Here we discuss two such classifiers, first we discuss about KNN or k-nearest neighbor. Then we discuss a classifier that we can train from the sample documents using probabilistic model, Naive Bayes.

5.1 K-Nearest Neighbor (KNN)

Nearest neighbor classifier is the simplest of all classification algorithms in supervised learning. This is a method of classifying patterns based on the class label of the closest training patterns in the feature space. The common algorithms used here are the nearest neighbor(NN) algorithm, the k-nearest neighbor(kNN) algorithm, and the modified k-nearest neighbor (mkNN) algorithm.These are non-parametric methods where no model is fitted using the training patterns.The accuracy using nearest neighbor classifiers is good. It is guaranteed to yield an error rate no worse than twice the Bayes error rate which is the optimal error rate.There is no training time required for this classifier. In other words, there is no design time for training the classifier.Every time a test pattern is to be classified, it has to be compared with all the training patterns, to find the closest pattern. This classification time

could be large if the training patterns are large in number or if the dimensionality of the patterns is high.

Nearest neighbor algorithm
If there are n patterns X_1, X_2, ..., X_n in the training data,X and a test pattern P, if X_k is the most similar pattern to P from X, then the class of P is the class of X_k. The similarity is usually measured by computing the distance from P to the patterns X_1, X_2, ..., X_n. If $d(P, X_i)$ is the distance from P to X_i, then P is the assigned the class label of X_k where

$$d(P, X_k) = mind(P, X_i)$$

where $i = 1, 2, \ldots, n$

k-Nearest Neighbor (kNN) classification algorithm An object is classified by a majority vote of the class of its neighbors.The object is assigned to the class most common amongst its k nearest neighbors. If $k = 1$, this becomes the nearest neighbor algorithm.This algorithm may give a more correct classification for boundary patterns than the NN algorithm. The value of k has to be specified by the user and the best choice depends on the data. Larger values of k reduce the effect of noise on the classification. The value of k can be arbitrary increased when the training data set is large in size.The k value can be chosen by using a validation set and choosing the k value giving best accuracy on the validation set. The main disadvantage of kNN algorithm is that it is very time consuming especially when the training data is large. To overcome this problem, a number of algorithms have been proposed to access the k nearest patterns as fast as possible. Modified k-Nearest neighbor (mkNN) classifier The contribution of the neighbors to the classification is weighted according to its distance from the test pattern. Hence, the nearest neighbor contributes more to the classification decision than the neighbors further away.One weighting scheme would be to give each neighbor a weight of $1/d$ where d is the distance from P to the neighbor.Another weighting scheme finds the weight from the neighbor as $w_i = (d_k - d_i)/(d_k - d_1)$ if $d_k \neq d_1$ else $w_i = 1$, where $i = 1, 2, \ldots, k$.

The value of w_i varies from 1 for the closest pattern to 0 for the farthest pattern among the k closest patterns.This modification would mean that outliers will not affect the classification as much as the kNN classifier.

5.2 Naive Bayes

A naive Bayes classifier is based on applying Bayes theorem to find the class of a unknown pattern. The assumption made here is that every feature is class conditionally independent. Due to this assumption, the probabilistic classifier is simple. In other words, it is assumed that the effect of each feature on a given class is independent of the value of other features. Since this simplifies the computation, though it may not be always true, it is considered to be a naive classifier. Even though this assumption is made, the Naive Bayes Classifier is found to give results comparable in performance to other classifiers like neural network classifiers and classification trees. Since the calculations are simple, this classifier can be used for large databases where the results are obtained fast with reasonable accuracy. Using the minimum error rate classifier, we classify the pattern X to the class with the maximum posterior probability $P(C|X)$.[5]

In text classification let us assume that t_1,t_2,t_3,...t_N are the N indexing terms of some unknown document D_i. The probability taht D_i belongs to class C_k(where C_k is one of the predefines categories) can be written as $P(C_k|t_1, t_2, .., t_N)$. Using Bayes theorem, this can be written as

$$P(C_k|t_1, t_2, t_3, ..., t_N) = \frac{P(C_k, t_1, t_2, t_3, ..., t_N)}{P(t_1, t_2, t_3, ..., t_N)}$$
$$P(C_k|t_1, t_2, t_3, ..., t_N) = \frac{P(C_k)P(t_1, t_2, t_3, ..., t_N|C_k)}{P(t_1, t_2, t_3, ..., t_N)}$$

In the above equation we are only interseted in the numerator of the fraction as the class value does not depend on the denominator. Since every feature t_i is independent of every other feature t_j , for ji, given the class C we can write:

$$P(t_i, t_j|C_k) = P(t_i|C_k)P(t_j|C_k)$$

So we get,

$$P(C_k, t_1, t_2, t_3, ..., t_N) = P(C)P(t_1|C_k)P(t_2|C_k)P(t_3|C_k)P(t_N|C_k)$$
$$P(C_k, t_1, t_2, t_3, ..., t_N) = P(C_k) * \prod_{i=1}^{N} P(t_i|C_k)$$

Under the above independence assumptions, the conditional distribution over the class variable C_k can be written as:

$$P(C_k|t_1, t_2, t_3, ..., t_N) = \frac{P(C_k) * \prod_{i=1}^{N} P(t_i|C_k)}{Z}$$

where $Z = P(t_1, t_2, t_3, ..., t_N)$ called the scaling factor which is constant when the terms are known.The Naive Bayes classification uses only the prior probabilities of classes $P(C_k)$ and the independent probability distributions $P(t_i|C_k)$.

Constructing a classifier from the probability model :
The discussion so far has derived the independent feature model, that is, the naive Bayes probability model. The naive Bayes classifier combines this model with a decision rule. One common rule is to pick the hypothesis that is most probable; this is known as the maximum a posteriori or MAP decision rule. The corresponding classifier, a Bayes classifier, is the function that assigns a class label $\hat{y} = C_k$ for some k as follows:

$$\hat{y} = \underset{k \in \{1,...,K\}}{\operatorname{argmax}} \; p(C_k) \prod_{i=1}^{n} p(x_i|C_k)$$

Chapter 6

Ontology

There are various aspects of interpreting the meaning of the word "ontology". Following [7] we can distinguish between the use as an uncountable noun,Ontology, with uppercase initial and the use as a countable noun ,an ontology, with lowercase initial. The uncountable noun "Ontology "refers to the branch of philosophy which deals with the nature and structure of reality. Aristotle dealt with this subject in his Metaphysics and defined Ontology as the science of being qua being, i.e., the study of attributes that belong to things because of their very nature. Unlike the experimental sciences, which aim at discovering and modeling reality under a certain perspective, Ontology focuses on the nature and structure of things per se, independently of any further considerations, and even independently of their actual existence. For example, it makes perfect sense to study the Ontology of unicorns and other fictitious entities: although they do not have actual existence, their nature and structure can be described in terms of general categories and relations. Whereas, in the second case, which reflects the most prevalent use in Computer Science, we refer to an ontology as a special kind of information object or computational artifact. According to [8, 9],the account of existence in this case is a pragmatic one: For AI systems, what exists is that which can be represented. According to wikipedia,we can define "in computer science and information science, an ontology is a formal naming and definition of the types, properties, and interrelationships of the entities that really or fundamentally exist for a particular domain of discourse". Computational ontologies are a means to formally model the structure of a system, i.e., the relevant entities and relations that emerge from its observation, and which are useful to our purposes. An example of such a system can be a company

with all its employees and their interrelationships. The ontology engineer analyzes relevant entities and organizes them into concepts and relations, being represented, respectively, by unary and binary predicates. The backbone of an ontology consists of a generalization/specialization hierarchy of concepts, i.e., a taxonomy. Supposing we are interested in aspects related to human resources, then Person, Manager, and Researcher might be relevant concepts, where the first is a superconcept of the latter two. Cooperates-with can be considered a relevant relation holding between persons. A concrete person working in a company would then be an instance of its corresponding concept.

In 1993, Gruber originally defined the notion of an ontology as an explicit specification of a conceptualization [8]. In 1997, Borst defined an ontology as a formal specification of a shared conceptualization [9]. This definition additionally required that the conceptualization should express a shared view between several parties, a consensus rather than an individual view. Also, such conceptualization should be expressed in a (formal) machine readable format. In 1998, Studer et al. [10] merged these two definitions stating that: An ontology is a formal, explicit specification of a shared conceptualization. All these definitions were assuming an informal notion of conceptualization .In the following, we shall revisit such discussion, by focusing on the three major aspects of the definition by Studer et al.:
i) Conceptualization
ii) Explicit specification.
iii) Meaning of shared .[1]
It is the task of this chapter to provide a concise view of these aspects in the following sections. We will use the examples from [1] to clarify the above aspects of ontology. It lies in the nature of such a chapter that we have tried to make it more precise and formal than many other useful definitions of ontologies that do exist but that do not clarify terms to the degree of accuracy that we target here.

6.1 Conceptualisation

Gruber [8, 9] refers to the notion of a conceptualization according to Genesereth and Nilsson [11], who claim: A body of formally represented knowledge is based on a conceptualization: the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that

hold among them. A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose. Every knowledge base, knowledge-based system, or knowledge-level agent is committed to some conceptualization, explicitly or implicitly. Despite the complex mental nature of the notion of conceptualization, Genesereth and Nilsson choose to explain it by using a very simple mathematical representation: an extensional relational structure.

6.1.1 Extensional relational structure

"An extensional relational structure, (or a conceptualization according to Genesereth and Nilsson), is a tuple (D, R) where D is a set called the universe of discourse and R is a set of relations on D. Note that, in the above definition, the members of the set R are ordinary mathematical relations on D, i.e., sets of ordered tuples of elements of D. So each element of R is an extensional relation, reflecting a specific world state involving the elements of D"[1], such as we describe in the following example.

Example. Let us consider human resources management in a large software company with 50,000 people, each one identified by a number (e.g., the social security number, or a similar code) preceded by the letter I. Let us assume that our universe of discourse D contains all these people, and that we are only interested in relations involving people. Our R will contain some unary relations, such as Person, Manager, and Researcher, as well as the binary relations reports-to and cooperates-with. The corresponding extensional relation structure (D, R) looks as follows:
$D = \{I000001, ..., I050000, ...\}$
$R = \{Person, Manager, Researcher, cooperates - with, reports - to\}$
Relation extensions reflect a specific world. Here, we assume that Person comprises the whole universe D and that Manager and Researcher are strict subsets of D. The binary relations reports-to and cooperates-with are sets of tuples that specify every hierarchical relationship and every collaboration in our company. Here, $I046758$, a researcher, reports to his manager $I034820$, and cooperates with another researcher, namely $I044443$.
$Person = D$
$Manager = \{..., I034820, ...\}$
$Researcher = \{..., I044443, ..., I046758, ...\}$
$reports - to = \{..., (I046758, I034820), (I044443, I034820), ...\}$

$cooperates - with = \{..., (I046758, I044443), ...\}$
Despite its simplicity, this extensional notion of a conceptualization does not really fit our needs and our intuition, mainly because it depends too much on a specific state of the world. Arguably, a conceptualization is about concepts. Now, should our concept of reports-to change when the hierarchical structure of our company changes? Indeed, as discussed in [**?**],a conceptualization should not change when the world changes. Otherwise, according to the Genesereth and Nilssons view given in above ,entioned definition, every specific people interaction graph, would correspond to a different conceptualization, as shown in this example.

Example. Let us consider the following alteration of the previous example with $D\prime = D$ and $R\prime =$ Person, Manager, Researcher, reports-to, cooperates-with where reports-to = reports-to (I034820, I050000). Although we only added one new reporting relationship, it is obvious that $(D, R) \neq (D\prime, R\prime)$ and, thus, we have two different conceptualizations according to Genesereth and Nilsson. The problem is that the extensional relations belonging to R reflect a specific world state. However, we need to focus on the meaning of the underlying concepts, which are independent of a single world state: for instance, the meaning of cooperates-with lies in the particular way two persons act in the company.

In practice, understanding such meaning implies having a rule to decide, observing different behavior patterns, whether or not two persons are cooperating. Suppose that, in our case, for two persons I046758 and I044443 to cooperate means that (1) both declare to have the same goal; (2) both do something to achieve this goal. Then, the meaning of cooperating can be defined as a function that, for each global behavioral context involving all our universe, gives us the list of couples who are actually cooperating in that context. The reverse of this function grounds the meaning of a concept in a specific world state. Generalizing this approach, and abstracting from time for the sake of simplicity, we shall say that an intensional relation (as opposed to an extensional relation) is a function from a set of maximal world states (the global behavioral contexts in our case) into extensional relations. This is the common way of expressing intensions, which goes back to Carnap and is adopted and extended in Montagues semantics. To formalize this notion of intensional relation, we first have to clarify what a world and a world state is. We shall define them with reference to the notion of system, which will be

given for granted: since we are dealing with computer representations of real phenomena, a system is simply the given piece of reality we want to model, which, at a given degree of granularity, is perceived by an observing agent (typically external to the system itself) by means of an array of observed variables. In our case, this system will be an actual group of people interacting in certain ways. For the sake of simplicity, we shall assume to observe this system at a granularity where single persons can be considered as atoms, so we shall abstract, e.g., from body parts. Moreover, we shall assume that the only observed variables are those which tell us whether a person has a certain goal (belonging to a predetermined list), and whether such person is actually acting to achieve such goal. Supposing there is just one goal, we have 50,000 + 50,000 = 100,000 variables. Each combination of such variables is a world state. Two different agents (outside the observed system) will share the same meaning of cooperating if, in presence of the same world states, will pick up the same couples as instances of the cooperates-with relation. If not, they will have different conceptualizations, i.e., different ways of interpreting their sensory data. For instance, an agent may assume that sharing a goal is enough for cooperating, while the other may require in addition some actual work aimed at achieving the goal.

6.1.2 Intensional relation, or conceptual relation

With respect to a specific system S we want to model, a world state for S is a maximal observable state of affairs, i.e., a unique assignment of values to all the observable variables that characterize the system. A world is a totally ordered set of world states, corresponding to the systems evolution in time. If we abstract from time for the sake of simplicity, a world state coincides with a world. At this point, we are ready to define the notion of an intensional relation in more formal terms, building on, as follows:

Definition (Intensional relation, or conceptual relation) :
"Let S be an arbitrary system, D an arbitrary set of distinguished elements of S, and W the set of world states for S (also called worlds, or possible worlds). The tuple $< D, W >$ is called a domain space for S, as it intuitively fixes the space of variability of the universe of discourse D with respect to the possible states of S. An intensional relation (or conceptual relation) ρ^{n} of arity n on $< D, W >$ is a total function $\rho^{\mathrm{n}} : W \rightarrow 2^{\mathrm{D^n}}$ from the set W into the set of all n-ary(extensional) relations on D. Once we have clarified

what a conceptual relation is, we give a representation of a conceptualization. Below, we show how the conceptualization of our human resources system looks like."

Definition (Intensional relational structure, or conceptualization)
"An intensional relational structure (or a conceptualization according to Guarino) is a triple $C = (D, W, R)$ with
D a universe of discourse
W a set of possible worlds
R a set of conceptual relations on the domain space $< D, W >$." Coming back to the previous examples, we can see them as describing two different worlds compatible with the following conceptualization C:
$D = \{I000001, ..., I050000, ...\}$ the universe of discourse
$W = \{w_1, w_2, ...\}$ the set of possible worlds
$R = \{Person_1, Manager_1, Researcher_1, cooperates - with_2, reports - to_2\}$ the set of conceptual relations For the sake of simplicity, we assume that the unary conceptual relations, viz., $Person_1, Manager_1$, and $Researcher_1$, are rigid, and, thus, map to the same extensions in every possible world. We do not make this specific assump- tion here for the binary $reports - to_2$ and $cooperates - with_2$:
for all worlds w in W : $Person_1(w) = D$
for all worlds w in W : $Manager_1(w) = \{..., I034820, ...\}$
for all worlds w in W : $Researcher_1(w) = \{..., I044443, ..., I046758, ...\}$
$reports - to_2(w_1) = \{..., (I046758, I034820), (I044443, I034820), , ...\}$
$reports - to_2(w_2) = \{..., (I046758, I034820), (I044443, I034820), (I034820, I050000), ...\}$
$reports - to_2(w_3) = ...$
$cooperates - with_2(w_1) = \{..., (I046758, I044443), ...\}$
$cooperates - with_2(w_2) = ..$

6.2 Explicit Specification

In practical applications, as well as in human communication, we need to use a language to refer to the elements of a conceptualization: for instance, to express the fact that I046758 cooperates with I044443, we have to introduce a specific symbol (formally, a predicate symbol, say cooperates-with, which, in the users intention, is intended to represent a certain conceptual relation. We say in this case that our language (let us call it L) commits to a con-

ceptualization. Suppose now that L is a first-order logical language, whose nonlogical symbols (i.e., its signature, or its vocabulary) are the elements of the set $I046758, I044443, cooperates - with, reports - to$. How can we make sure that such symbols are interpreted according to the conceptualization we commit to? For instance, how can we make sure that, for somebody who does not understand English, cooperates-with is not interpreted as corresponding to our conceptualization of reports-to, and vice versa? Technically, the problem is that a logical signature can, of course, be interpreted in arbitrarily many different ways. Even if we fix a priori our interpretation domain (the domain of discourse) to be a subset of our cognitive domain, the possible interpretation functions mapping predicate symbols into proper subsets of the domain of discourse are still unconstrained. In other words, once we commit to a certain conceptualization, we have to make sure to only admit those models which are intended according to the conceptualization. For instance, the intended models of the cooperates-with predicate will be those such that the interpretation of the predicate returns one of the various possible extensions (one for each possible world) of the conceptual relation denoted by the predicate. The problem however is that, to specify what such possible extensions are, we need to explicitly specify our conceptualization, while conceptualizations are typically in the mind of people, i.e., implicit. Here emerges the role of ontologies as explicit specifications of conceptualizations.

In principle, we can explicitly specify a conceptualization in two ways: extensionally and intensionally. In our example, an extensional specification of our conceptualization would require listing the extensions of every (conceptual) relation for all possible worlds. However, this is impossible in most cases (e.g., if the universe of discourse D or the set of possible worlds Ware infinite) or at least very impractical. In our running example, we are dealing with thousands of employees and their possible cooperations can probablynot be fully enumerated. Still, in some cases it makes sense to partially specify a conceptualization in an extensional way, by means of examples, listing the extensions of conceptual relations in correspondence of selected, stereotypical world states. In general, however, a more effective way to specify a conceptualization is to fix a language we want to use to talk of it, and to constrain the interpretations of such a language in an intensional way, by means of suitable axioms (called meaning postulates). For example, we can write simple axioms stating that reports-to is asymmetric and intransitive, while cooperates-with is symmetric, irreflexive, and intransitive. In short, an

ontology is just a set of such axioms, i.e., a logical theory designed in order to capture the intended models corresponding to a certain conceptualization and to exclude the unintended ones. The result will be an approximate specification of a conceptualization: the better intended models will be captured and non-intended models will be excluded. The axioms for intensionally and explicitly specifying the conceptualization can be given in an informal or formal language L. As explained in the introduction, [10] requires that the explicit specification must be formal in addition to what proposed in the definitions in [7, 9]. Formal refers to the fact that the expressions must be machine readable, hence natural language is excluded. Let us now discuss all the notions above in a more formal way.

6.3 Committing to a Conceptualisation

Let us assume that our language L is (a variant of) a first-order logical language, with a vocabulary V consisting of a set of constant and predicate symbols (we shall not consider function symbols here). We shall introduce the notion of ontological commitment by extending the standard notion of a (extensional) first order structure to that of an intensional first order structure.

6.3.1 Extensional first-order structure

"Let L be a first order logical language with vocabulary V and $S = (D, R)$ an extensional relational structure. An extensional first order structure (also called model for L) is a tuple $M = (S, I)$, where I (called extensional interpretation function) is a total function $I : V \rightarrow D \cup R$ that maps each vocabulary symbol of V to either an element of D or an extensional relation belonging to the set R."

6.3.2 Intensional first-order structure

"It is also known as ontological commitment. Let L be a first-order logical language with vocabulary V and $C = (D, W, R)$ an intensional relational structure (i.e., a conceptualization). An intensional first order structure (also called ontological commitment) for L is a tuple $K = (C, I)$, where I (called intensional interpretation function) is a total function $I : V \rightarrow D \cup R$ that

maps each vocabulary symbol of V to either an element of D or an intensional relation belonging to the set R."

It should be clear now that the definition of ontological commitment extends the usual (extensional) definition of meaning for vocabulary symbols to the intensional case, substituting the notion of model with the notion of conceptualization. Example : Coming back to our previous example, the vocabulary V coincides with the relation symbols, i.e., $V = \{Person, Manager, Researcher, reports-to, cooperates-with\}$

Our ontological commitment consists of mapping the relation symbol $Person$ to the conceptual relation $Person_1$ and proceeding alike with *Manager*, *Researcher*, *reports-to*, and *cooperates-with*.

6.3.3 Specifying a Conceptualization

As we have seen, the notion of ontological commitment is an extension of the standard notion of model. The latter is an extensional account of meaning, the former is an intensional account of meaning. But what is the relationship between the two? Of course, once we specify the intensional mening of a vocabulary through its ontological commitment, somehow we also constrain its models. Let us introduce the notion of intended model with respect to a certain ontological commitment for this purpose.

Definition(Intended models)
Let $C = (D, W, R)$ be a conceptualization, L a first-order logical language with vocabulary V and ontological commitment $K = (C, I)$. A model $M = (S, I)$, with $S = (D, R)$, is called an intended model of L according to K if and only if
1. For all constant symbols $c \in V$ we have $I(c) = I(c)$.
2. There exists a world $w \in W$ such that, for each predicate symbol $v \in V$.
3. There exists an intensional relation $\rho \in R$ such that $I(v) = \rho$ and $I(v) = \rho(w)$.
The set $I_K(L)$ of all models of L that are compatible with K is called the set of intended models of L according to K. Condition 1 above just requires that the mapping of constant symbols to elements of the universe of discourse is identical. The first example does not introduce any constant symbols. Condition 2 states that there must exist a world such that ev-

ery predicate symbol is mapped into an intensional re- lation whose value, for that world, coincides with the extensional interpretation of such symbol. This means that our intended model will be so to speak a description of that world. In the next example, for instance, we have that, for w_1 , $I(Person) = \{I000001, ..., I050000, ...\} = Person_1(w_1)$ and $I(reports - to) = \{..., (I046758, I034820), (I044443, I034820), (I034820, I050000), ...\} = reports - to_2(w_1)$. With the notion of intended models at hand, we can now clarify the role of an ontology, considered as a logical theory designed to account for the intended meaning of the vocabulary used by a logical language. In the following, we also provide an ontology for our running example.

Definition (Ontology) "Let C be a conceptualization, and L a logical language with vocabulary V and ontological commitment K. An ontology O_K for C with vocabulary V and ontological commitment K is a logical theory consisting of a set of formulas of L, designed so that the set of its models approximates as well as possible the set of intended models of L according to K".
In the following we build an ontology O consisting of a set of logical formulae. Through O_1 to O_6 we specify our human resources domain with increasing precision.

Taxonomic Information : We start our formalization by specifying that Researcher and Manager are sub-concepts of Person:
$O_1 = \{Researcher(x) \rightarrow Person(x), Manager(x) \rightarrow Person(x)\}$
Domains and Ranges : We continue by adding formulae to O_1 which specify the domains and ranges of the binary relations:
$O_2 = O_1 \cup \{cooperates - with(x, y) \rightarrow Person(x) \cap Person(y),\ reports - to(x, y) \rightarrow Person(x) \cap Person(y)\}$
Symmetry : cooperates-with can be considered a symmetric relation:
$O_3 = O_2 \cup \{cooperates - with(x, y) \iff cooperates - with(y, x)\}$
Transitivity : Although arguable, we specify reports-to as a transitive relation:
$O_4 = O_3 \cup \{reports - to(x, z) \leftarrow reports - to(x, y) \cap reports - to(y, z)\}$
Disjointness : There is no Person who is both a Researcher and a Manager:
$O_5 = O_4 \cup \{Manager(x) \rightarrow \neg Researcher(x)\}$

6.4 Ontology - A simple Perspective

We can define ontology by a mathematical structure. A core ontology is a structure

$$O = (C, \preceq_C, R, \sigma, \preceq_R)$$

consisting of two disjoint sets C and R whose elements are called concept identifiers and relation identifiers respectively, a partial order $\preceq_C$ on C, called concept hierarchy or taxonomy, a function $\sigma : R \to C^+$ called signature, a partial order $\preceq_R$ on R, called relation hierarchy, where $r_1 \preceq_R r_2$ implies $|\sigma(r_1)| = |\sigma(r_2)|$ and $\prod_i(\sigma(r_1)) \preceq_C \prod_i(\sigma(r_2))$, for each $1 \leq i \leq |\sigma(r_1)|$ and C^+ is the set of tuples over C with at least one element and $\prod_i$ is the i-th component of a given tuple.
$Definition(SubconceptsandSuperconcepts)$
If $c_1 \prec_C c_2$ for any $c_1, c_2 \in C$, then c_1 is a subconcept (specialization) of c_2 and c_2 is a superconcept (generalization) of c_1 . If $c_1 \prec_C c_2$ and there exists no $c_3 \in C$ with $c_1 \prec_C c_3 \prec_C c_2$, then c_1 is a direct subconcept of c_2 , and c_2 is a direct superconcept of c_1 , denoted by $c_1 \prec c_2$.

The partial order $\preceq_C$ relates the concepts in an ontology in form of specialization and generalization relationships, resulting in a hierarchical arrangement of concepts. These relationships correspond to what is generally known as is-a or is-a-special-kind-of relations. Often we will call concept identifiers and relation identifiers just concepts and relations, respectively, for sake of simplicity. Almost all relations in practical use are binary. For those relations, we define their domain and their range.

Definition (Domain and Range)
For a relation $r \in R$ with $|\sigma(r)| = 2$, we define its domain and its range by $dom(r) = \prod_1(\sigma(r))$ and $range(r) = \prod_2(\sigma(r))$.

According to the international standard ISO 704, we provide names for the concepts (and relations). Instead of name, we here call them sign or lexical entries to better describe the functions for which they are used.

Definition (Lexicon for an Ontology) A lexicon for an ontology O is a tuple $Lex = (S_C, Ref_C)$ consisting of a set S_C , whose elements are called signs for concepts (symbols), and a relation $Ref_C \subset S_C X C$ called lexical reference

for concepts, where $(c, c) \in Ref_C$ holds for all $c \in C \cap S_C$. Based on Ref_C , for $s \in S_C$ we define $Ref_C(s) = \{c \in C | (s, c) \in Ref_C\}$. Analogously, for $c \in C$ it is ${Ref_C}^1(c) = \{s \in S_C | (s, c) \in Ref_C\}$. An ontology with lexicon is a pair (O, Lex) where O is an ontology and Lex is a lexicon for O.

Chapter 7

Use of Ontology in Text Classification

The drawback of conventional system is each document is expressed as a term vector where each terms is assumed to be independent of each other. However it is not the obvious case and we are ignoring the interrelation between the terms that can also affect the classification task. Using ontology we will be able to address this problem and come up witth a solution.

As we already discussed that an ontology O can be represented by a structure $(C, \preceq_C, R, \sigma, \preceq_R)$ where C denotes the set of concepts, R denotes the set of relations, $\preceq_C$ is the partial order that shows the hierarchy of concepts or taxonomy, $\preceq_R$ is the partial order that shows the hierarchy of relations and a function $\sigma : R \rightarrow C^+$ that shows relationships between various concepts. In our approach, we exploit this background knowledge about concepts that is explicitly given according to our ontological model.

Using Vector Space Model, we describe each document as a term vector. Say for example, we assume that after preprocessing step we have N number of terms and we have a document set with M number of documents. Here we describe each document $d_i \in D$ as N-dimensional vector, so $d_i = (tf(d_i, t_1), tf(d_i, t_2), \ldots, tf(d_i, t_N))$. Now to use the ontological knowledge, we extend each term vector t_j by new entries for ontological concepts c appearing in the document set D. We would have already built the ontology structure O from document set D and a lexicon for that ontology. It should be noted that a lexicon is a tuple $Lex = (S_C, Ref_C)$

consisting of a set S_C , whose elements are called signs for concepts (symbols), and a relation $Ref_C \subset S_C X C$ called lexical reference for concepts, where $(c, c) \in Ref_C$ holds for all $c \in C \cap S_C$ and for $s \in S_C$ we define $Ref_C(s) = c \in C|(s, c) \in Ref_C$. So we can replace all the terms t_j with its corresponding Ref_C values and we get a new vector for each document d_i, the concept vector $(cf(d_i, c_1), cf(d_i, c_2), \ldots, cf(d_i, c_l))$ where l is the cardinality of the set C and $cf(d_i, c)$ denotes the frequency of the appearance of concept $c \in C$ in document d_i as indicated by applying the reference function Ref_C to all terms in the document set D.

After getting the concept vector for each document $d_i \in D$, we can have two different strategies to be made. The first one is to replace the term vector with the concept vector for each document in the document set. Or we can concatenate the concept vector with the corresponding term vector for each document. Thus in the first case, for each $d_i \in D$, we will have $d_i = (cf(d_i, c_1), cf(d_i, c_2), \ldots, cf(d_i, c_l))$ and each N-dimensional vector is relaced by a l-dimensional vector. Whereas in the second case, each document $d_i \in D$ will be represented by

$$d_i = (tf(d_i, t_1), tf(d_i, t_2), \ldots, tf(d_i, t_N), (cf(d_i, c_1), cf(d_i, c_2), \ldots, cf(d_i, c_l))$$

Hence, each document is now represented as $(N + l)$-dimensional vector. A term that also appears in the ontology would be accounted for at least twice in the new vector representation, i. e., once as part of the old term vector and at least once as part of the new concept vector . It could be accounted for also more often, because a term like bank has several corresponding concepts in the ontology.

To extract the concepts from texts, we can develop a detailed process, that can be used with any ontology with lexicon. The overall process comprises five processing steps that are described in the following.

7.1 Candidate Term Detection

Due to the existence of multi-word expressions, the mapping of terms to concepts can not be accomplished by querying the lexicon directly for the single words in the document. We can use several candidate term detection algorithm as discussed in [12] that builds on the basic assumption that

finding the longest multi-word expressions that appear in the text and the lexicon will lead to a mapping to the most specific concepts. The algorithm works by moving a window over the input text, analyzing the window content and either decreasing the window size if unsuccessful or moving the window further. For English, a window size of 4 is sufficient to detect virtually all multi-word expressions.

7.2 Syntactical Patterns

Querying the lexicon directly for any expression in the window will result in many unnecessary searches and thereby in high computational requirements. Luckily, unnecessary search queries can be identified and avoided through an analysis of the part-of-speech (POS) tags of the words contained in the current window. Concepts are typically symbolized in texts within noun phrases. By defining appropriate POS patterns and matching the window content against these, multi-word combinations that will surely not symbolize concepts can be excluded in the first hand and different syntactic categories can be disambiguated.

7.3 Morphological Transformations

Typically the lexicon will not contain all inflected forms of its entries. If the lexicon interface or separate software modules are capable of performing base form reduction on the submitted query string, queries can be processed directly. For example, this is the case with WordNet. If the lexicon, as in most cases, does not contain such functionalities, a simple fallback strategy can be applied. Here, a separate index of stemmed forms is maintained. If a first query for the inflected forms on the original lexicon turned out unsuccessful, a second query for the stemmed expression is performed.

7.4 Word Sense Disambiguation

Having detected a lexical entry for an expression, this does not necessarily imply a one-to-one mapping to a concept in the ontology. Although multi-word-expression support and POS pattern matching reduce ambiguity, there may arise the need to disambiguate an expression versus multiple possible

concepts. The word sense disambiguation (WSD) task is a problem in its own right [14] and was not the focus of this paper.

7.5 Generalization

The last step in the process is about going from the specific concepts found in the text to more general concept representations. However, we do not only add the concepts directly representing the terms but also the corresponding superconcept along the path to the root of the concept hierarchy. An important issue here is to restrict the number of levels up in the hierarchy considered for adding superconcepts. The following procedure realizes this idea by adding to the concept frequency of higher level concepts in a document d_i the frequencies of their subconcepts (of at most r levels down in the hierarchy). The vectors we consider are first of the form $d_i = (tf(d_i, t_1), tf(d_i, t_2), \ldots, tf(d_i, t_N), (cf(d_i, c_1), cf(d_i, c_2), \ldots, cf(d_i, c_l))$(the concatenation of an initial term vector representation with a concept vector). Then the frequencies of the concept vector part are updated, for a user-defined $r \in N$, in the following way. For all $c \in C$, replace $cf(d_i, c)$ by $cf\prime(d_i, c) = \sum_{b \in H(c,r)} cf(d_i, b)$ where

$$H(c, r) = \{c\prime | \exists c_1, c_2, \ldots, c_i \in C : c\prime \prec c_1 \prec c_2 \prec \cdots \prec c_i = c, 0 \leq i \leq r\}$$

gives for a given concept c the r next subconcepts in the taxonomy. In particular $H(c, \infty)$ returns all subconcepts of c. This implies that the strategy $r = 0$ does not change the given concept frequencies, $r = l$ adds to each concept the frequency counts of all subconcepts in the l levels below it in the ontology and $r = \infty$ adds to each concept the frequency counts of all its subconcepts [13].

Chapter 8

Practical Implementation

To implement all the above-mentioned ideas, we use 'Apache Lucene' which is a high-performance, full-featured text search engine library written entirely in Java. It is a technology suitable for nearly any application that requires full-text search, especially cross-platform. In the next section we will discuss in details how Lucene implements all those ideas. In our experiment we use the most widely used test collection for text categorization research, Reuters-21578. To capture the ontological(it is better to say taxonomical, as we keep our focus only on the hierarchy of the concepts) information, we have used Wordnet 3.1.

Our first step is to index each document of the document set. Our indexing process consists of four steps, 1.Structure analysis and tokenization ; 2.Stopword removal ; 3.Morphological normalization ; 4.Weighting. Lucene provides appropriate classes for carrying out all the steps. We first tokenize the document and then convert it into lowercase. We keep a stopword list and remove all those stopwords. Then we normalise the words into its stem form. For weighting, Lucene uses a simple tf-idf formula where the term frequency of term t_j in document in d_i is measured as $tf(d_i, t_j) = \sqrt{frequency}$ where frequency is the number of appearance of the term t_j in document d_i and inverse document frequency idf is measured by the formula $idf(t_j)$ $= 1 + log(|D|/(docfreq + 1))$ where $|D|$ is the number of documents in the document set D and docfreq is the number of documents where the term t_j appears. One important fact to mention about Lucene is that it uses inverse index, i.e. we don't think of a document as consisting of a number of terms. Instead each term will have its corresponding list of documents where the

term appears.

However the most useful part in the indexing process is an additional step that maps all the terms with Wordnet. Here we exploit the taxonomical relation that is available in Wordnet. For each term, we replace it with the corresponding leading term from the Wordnet. It can be thought of as replacing the term vector with the concept vector as we described in chapter 7. It makes the difference from the conventional process.

As we are not interested to compare the performance of the classifiers, in this paper, we only use K-Nearest neighbor classifier with $k = 18$. It should be noted that we could get better accuracy using some other classifier, however we are only interested to show the contrast in performance that is made using ontology.

Chapter 9

Experimental Result

As it is already mentioned, we have used the test collection of Reuters-21578 in our experiment. It consists of 11413 training documents and 4024 testing documents and 105 different classes. From this dataset, we choose a training set of 4332 documents and 1676 documents for testing purpose, with 14 different classes. The experimental result has been shown by a confusion matrix. First, the confusion matrix of these 14 classes has been shown when the ontological information is not used.The confusion matrices are constructed as follows, the rows represent the documents belonging to that class and the columns represent the class in which the documents are classified. From this matrix we can calculate the accuracy rate of the classification task. The first table is constructed without using the ontological information and then we classify the documents using ontological information which is shown in the second table.

Class	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)
acq(1)	596	0	1	0	0	9	0	0	1	2	1	2	2	0
bop(2)	0	19	0	0	0	0	1	9	0	0	0	0	0	0
coffee(3)	0	2	21	0	0	1	0	0	0	3	0	0	0	0
corn(4)	1	1	0	0	0	1	0	0	0	45	0	0	1	0
cpi(5)	0	1	0	0	11	2	0	8	0	0	2	0	3	1
crude(6)	4	0	0	0	0	159	0	0	0	4	0	0	1	0
dlr(7)	0	0	0	0	0	0	7	1	0	0	2	0	20	1
gnp(8)	0	2	0	0	0	1	0	28	0	0	2	0	1	0
gold(9)	3	0	0	0	0	0	0	0	22	0	2	0	1	0
grain(10)	1	1	1	1	0	2	0	1	1	126	0	1	1	0
inter(11)	0	0	0	0	0	0	0	0	0	1	55	0	53	2
ls(12)	0	0	0	0	0	0	0	0	1	10	0	13	0	0
m-fx(13)	1	0	0	0	0	0	7	3	1	0	10	0	118	3
m-sup(14)	0	0	0	0	0	0	0	1	0	0	6	0	3	15

So without using ontology we get an accuracy rate = 0.817952 and an error rate = 0.182048 for these 2763 documents and 14 classes. Before we discuss the various entries, we show the confusion matrix after using the ontological information using Wordnet using those same 2763 number of documents and 14 classes in same order.

Class	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)
acq(1)	630	0	2	0	0	11	2	0	0	8	4	2	11	1
bop(2)	0	13	0	0	0	0	0	5	0	0	2	0	9	0
coffee(3)	0	0	26	0	0	0	0	0	0	2	0	0	0	0
corn(4)	0	0	0	0	0	0	0	0	0	51	0	0	4	0
cpi(5)	0	0	1	0	7	3	0	10	0	0	3	0	3	1
crude(6)	12	0	0	0	0	169	0	0	0	2	1	0	1	0
dlr(7)	1	0	0	0	0	0	7	1	0	0	2	0	32	1
gnp(8)	0	1	0	0	0	2	0	17	0	0	7	0	8	0
gold(9)	5	0	1	0	0	3	0	0	20	0	0	0	1	0
grain(10)	1	0	0	1	1	2	1	0	0	129	3	2	8	0
inter(11)	4	0	0	0	0	0	0	1	0	0	73	0	47	5
ls(12)	2	0	0	0	0	2	0	0	0	7	0	8	5	0
m-fx(13)	7	0	0	0	0	3	8	1	0	0	24	0	125	4
m-sup(14)	0	1	0	0	0	0	0	1	0	1	9	0	1	19

After using ontology, we get an accuracy rate = 0.832429 and an error rate = 0.167571.If we look carefully on the above matrices, we can easily understand that in both the cases the diagonal elements are larger than other elements as expected as a diagonal element represents the true positive value that is the number of documents that the classifier identifies correctly. So we can easily make out that the performance in both cases are quite good. However our aim is to make the system better using ontology. If we compare the corresponding entries in both matrices, it will be clear that the ontology has outperformed the conventional system. From these tables we can claim the increase in accuracy in document classification after using ontology.

Chapter 10

Conclusion

In this paper we have discussed how we can exploit the relation of various terms present in a textual document by building an ontology structure. We used Wordnet to achieve this. Initially we have followed the conventional bag of words model where each document was represented as term vector and then we created the concept vector from the ontology for those documents. At last we used K-Nearest neighbor classifier to classify the documents when only term vectors were used, i.e. without using ontology and when the concatenation of term vectors and concept vectors were used, i.e. using the background knowledge from ontology. We found that using ontology the accuracy of the classification task has been improved over the conventional model. This is because expressing each document as a term vector only we are ignoring the interrelation between those terms.

However one thing that should be noted is that the ambiguity in our text collection is not known to us before classification. The improvement in performance of classification task using ontology will depend on the ambiguity present in the text collection. So we cant be too sure about a significant improvement in accuracy of document classification. Also we will have to face a trade-off between classification time and accuracy, because using ontology the accuracy is improved but at the same time it also takes more time.

Chapter 11

Scope of Improvement

Our experiment could have been done better in several ways. First of all, the choice of Wordnet to exploit the information of ontology should be inspected. As it is mentioned that we are only using the taxonomical information using the Wordnet. Instead of using Wordnet, we could create our own collection specific ontology in semi-automatic way and we could use that in our experiment. Though it would be a tedious job, the accuracy would increase quite significantly. Depending on the collection, the ontology structure will also change accordingly. Thus it would not only increase the accuracy, it would decrease the time taken for classification task. There are number of tools are now available that can be used to create the ontology semi-automatically and to express the ontology.

Another point should be mentioned, that is the use of K-Nearest neighbor as the only classifier. We already mentioned that in this paper we were only interested in the difference in performance due to the use of ontology, so we used the simplest classifier K-nearest neighbor in both the cases. However, it is seen that support vector machine(SVM) works most efficiently in these cases. Though it is a binary classifier, but we can use it number of times and get much higher accuracy than other classifier.

In our future work, we will try to build the ontology for our text collection in semi-automatic way and will choose support vector machine as the classifier to get higher accuracy . Moreover, we will try to implement ontology based text classification for other languages.

Bibliography

[1] S. Staab and R. Studer, editors. *Handbook on Ontologies*. Springer, 2004

[2] Nitin Indurkhya and FRED J. DAMERAU, editors. *Handbook of Natural Language Processing*. CRC Press, A Chapman and Hall Book.

[3] A. Hotho, S. Staab, and G. Stumme. *Explaining text clustering results using semantic structures*. In Principles of Data Mining and Knowledge Discovery, 7th European Conference, PKDD 2003, 2003.

[4] A. Hotho, S. Staab, and G. Stumme. *Ontologies improve text document clustering*. In Proc. of the ICDM 03, The 2003 IEEE International Conference on Data Mining, pages 541544, 2003.

[5] Peter Jackson and Isabelle Moulinier. *Natural Language Processing for Online Applications : Text Retrieval, Extraction and Categorization*. John Benjamins Publishing Company, Amsterdam / Philadelphia.

[6] Kobsa A (1993). *User modeling: recent work, prospects and hazards*. Schneider-Hufschmidt M, Khme T, Malinowski U (ed.) Adaptive User Interfaces: Principles and Practice. Elsevier Amsterdam.

[7] Gerhart A (2002) Open directory project search results and ODP status. Search Engine Guide.

[8] Guarino N, Giaretta P (1995) Ontologies and knowledge bases: towards a terminological clarification. In Mars N (ed.) Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing. IOS Press, Amsterdam, pp. 2532

[9] Balabanovic M, Shoham Y (1997).Fab: Content-based, collaborative recommendation. Communications of the ACM 40(3):6772.

[10] Middleton SE, Alani H, Shadbolt NR, De Roure DC (2002). *Exploiting synergy between ontologies and recommender systems. In International Workshop on the Semantic Web*. Proceedings of the 11th International World Wide Web Conference WWW-2002, Hawaii, USA.

[11] Felfernig A, Friedrich G, Jannach D, Zanker M (2006) *An integrated environment for the development of knowledge-based recommender applications*. International Journal of Electronic Commerce 11(2):1134.

[12] S. Bloehdorn and A. Hotho. *Boosting for Text Classification with Semantic Features*. In Proceedings of the MSW 2004 workshop at the 10th ACM SIGKDD Conference on Knowledge Discovery and Data Mining , 2004.

[13] S. Bloehdorn and P. Cimiano and A. Hotho and S.Staab. *An Ontology-based Framework for Text Mining*. Institute AIFB, University of Karlsruhe.

[14] N. Ide and J. Veronis. *Introduction to the special issue on word sense disambiguation: The state of the art*. Computational Linguistics, 24(1):140, 1998.