Non-Learning Semantic Analysis for Context Discovery and Sentiment Estimation: Transportation Application

Himanshu Verma

University of Nevada, Las Vegas, superbhimanshu@gmail.com

Follow this and additional works at: http://digitalscholarship.unlv.edu/thesesdissertations

Part of the Computer Engineering Commons, Computer Sciences Commons, and the Electrical and Computer Engineering Commons

Repository Citation

http://digitalscholarship.unlv.edu/thesesdissertations/2225

This Thesis is brought to you for free and open access by Digital Scholarship@UNLV. It has been accepted for inclusion in UNLV Theses, Dissertations, Professional Papers, and Capstones by an authorized administrator of Digital Scholarship@UNLV. For more information, please contact digitalscholarship@unlv.edu.
NON-LEARNING SEMANTIC ANALYSIS FOR CONTEXT DISCOVERY AND
SENTIMENT ESTIMATION: TRANSPORTATION APPLICATION

by

HIMANSHU VERMA

Bachelor of Technology
Indian Institute of Technology, Guwahati, India
2009

A thesis submitted in partial fulfillment
of the requirements for the

Master of Science in Engineering - Electrical Engineering

Department of Electrical and Computer Engineering
Howard R. Hughes College of Engineering
The Graduate College

University of Nevada, Las Vegas
August 2014
THE GRADUATE COLLEGE

We recommend the thesis prepared under our supervision by

Himanshu Verma

entitled

Non-Learning Analysis for Context Discovery and Sentiment Estimation: Transportation Application

is approved in partial fulfillment of the requirements for the degree of

Master of Science in Engineering – Electrical Engineering
Department of Electrical and Computer Engineering

Pushkin Kachroo, Ph.D., Committee Chair
Emma Regentova, Ph.D., Committee Member
Mei Yang, Ph.D., Committee Member
Haroon Stephen, Ph.D., Graduate College Representative
Kathryn Hausbeck Korgan, Ph.D., Interim Dean of the Graduate College

August 2014
ABSTRACT

Non-Learning Semantic Analysis for Context Discovery and Sentiment Estimation: Transportation Application

by

Himanshu Verma

⟨Dr. Pushkin Kachroo⟩, Examination Committee Chair
Professor of Electrical and Computer Engineering
University of Nevada, Las Vegas

With enormous amount of linguistic data present on web, text analysis has become one of the major fields of interest today. This field includes sentiment analysis, information retrieval, text document classification, knowledge based modeling, content similarity measure, data clustering, words prediction/correction, decision making etc. Managing and processing such data has vital importance. The field being quite broad, our focus is mainly on transportation related social media(Twitter) data extraction, text categorization/classification which can be further sub-divided into concept discovery, word sense disambiguation and sentiment analysis to analyze performance of existing transportation system worldwide. Concept discovery is the method of extracting the actual concept/context in which the text is about. This also allows us to filter irrelevant data. Word sense disambiguation is to find the correct sense in which a word is being used in a sentence. It is the basic necessity for concept discovery.
A lot of research has been done in this field with major improvements. However, when it comes to short texts, the field still seems in nascent stage. Moreover, most of the methods today require huge amount training corpus(database). Arranging such corpus is a cumbersome task and requires a lot of human effort. The other problem with the existing methods are that they require a set of defined concepts from which a concept is chosen and labeled to text. We will consider the case of finding a general context. In this work a novel approach has been proposed for word sense disambiguation which in turn allows us to find general context. For this purpose, I have used the existing knowledge based semantic dictionary called WordNet. This methodology helps in avoiding the use of huge corpus and works for general context recognition. Our focus is on short-texts(Tweets) but the concept is easily applicable to text documents as well.

Sentiment measuring technique was applied on extracted data and the scores were mapped to google maps based on the location information present in the tweets. This clearly points out the locations where people are more frustrated with the existing transportation system and need immediate attention for improvements.
ACKNOWLEDGEMENTS

I would like to thank Dr. Pushkin Kachroo for providing me the opportunity to work under his precious guidance. I would like to thank my family and friends who motivated me to work hard enough during this journey.
# TABLE OF CONTENTS

**ABSTRACT** iii

**ACKNOWLEDGEMENTS** v

**LIST OF FIGURES** viii

## 1 FRAMEWORK

1.1 Summary .................................................. 1
1.2 Motivation ................................................ 1
1.3 Problem Statement ........................................ 4
  1.3.1 Objective ............................................. 6
  1.3.2 Limitations of existing Text Analysis Systems ........ 6
  1.3.3 Solution .............................................. 7

## 2 LITERATURE SURVEY

2.1 Summary .................................................. 9
2.2 Introduction .............................................. 10
2.3 History of Classification .................................. 11
  2.3.1 Categorization of Classification Based on Input Space . 11
  2.3.2 Traditional Classification ................................ 12
  2.3.3 Representation of Text .................................. 13
  2.3.4 Features Selection ...................................... 14
  2.3.5 Definition of Learning ................................... 17
  2.3.6 Types of Classification Problems ......................... 19
2.4 Learning Classifiers ....................................... 20
  2.4.1 Naive Bayesian Classifier ............................... 20
  2.4.2 Rocchio Algorithm ..................................... 22
  2.4.3 K-Nearest Neighbour Algorithm ......................... 24
  2.4.4 Support Vector Machines(SVM) ......................... 24
  2.4.5 Categorization Based on Content Similarity ............ 27
  2.4.6 General Context Recognition- Short Text ................ 30
2.5 Conclusion ............................................... 32

## 3 WORDNET AND METHODS

3.1 Summary .................................................. 33
3.2 WordNet .................................................. 33
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Performance Analysis System 1</td>
<td>5</td>
</tr>
<tr>
<td>1.2</td>
<td>Performance Analysis System 2</td>
<td>6</td>
</tr>
<tr>
<td>2.1</td>
<td>Qualitative Analysis</td>
<td>9</td>
</tr>
<tr>
<td>2.2</td>
<td>Traditional Classification Flow</td>
<td>12</td>
</tr>
<tr>
<td>2.3</td>
<td>Machine Learning</td>
<td>18</td>
</tr>
<tr>
<td>2.4</td>
<td>Support Vector Machine</td>
<td>25</td>
</tr>
<tr>
<td>3.1</td>
<td>Example of WordNet Network</td>
<td>35</td>
</tr>
<tr>
<td>4.1</td>
<td>Context Determination Model for Documents</td>
<td>43</td>
</tr>
<tr>
<td>4.2</td>
<td>Context Determination Model for a Sentence</td>
<td>44</td>
</tr>
<tr>
<td>4.3</td>
<td>Synsets Network</td>
<td>47</td>
</tr>
<tr>
<td>4.4</td>
<td>Similarity Measure</td>
<td>48</td>
</tr>
<tr>
<td>5.1</td>
<td>Sentiment Analysis Flow</td>
<td>54</td>
</tr>
<tr>
<td>6.1</td>
<td>Similarity Measure 1</td>
<td>57</td>
</tr>
<tr>
<td>6.2</td>
<td>Similarity Measure 2</td>
<td>57</td>
</tr>
<tr>
<td>6.3</td>
<td>Proof of Concept</td>
<td>59</td>
</tr>
<tr>
<td>6.4</td>
<td>Focus Word: Traffic</td>
<td>61</td>
</tr>
<tr>
<td>6.5</td>
<td>Focus Word: Turn</td>
<td>61</td>
</tr>
<tr>
<td>6.6</td>
<td>Focus Word: State</td>
<td>61</td>
</tr>
<tr>
<td>6.7</td>
<td>Focus Word: Draw</td>
<td>62</td>
</tr>
<tr>
<td>6.8</td>
<td>Focus Word: Film</td>
<td>62</td>
</tr>
<tr>
<td>6.9</td>
<td>Sentiment Analysis Result Around the Globe</td>
<td>63</td>
</tr>
<tr>
<td>6.10</td>
<td>Sentiment Analysis Result Around the US</td>
<td>63</td>
</tr>
</tbody>
</table>
CHAPTER 1

FRAMEWORK

1.1 Summary

We have divided our problem statement into various sub-problems. This chapter provides the detailed description of the problem, issues with the existing solutions and then briefly describes the solution we have proposed. It provides macro level building blocks of the work presented in this document. This would be helpful for reader to have better understanding of the road map and overall framework.

1.2 Motivation

Thousands of people die every year in United States due to unfortunate accidents and crashes. Some of the facts about death tolls due to crashes are described below. This is one of the major concern which needs to be addressed by any means. Like most of the other fields technology comes as the savior for this purpose also. And the way we can handle such situations are by making our transportation system intelligent and smarter. In research world, this field is known as Intelligent Transportation System (ITS). A lot of research is going on in several top-notch universities for this purpose [30] [31] [32].
1. In 2007, 47.5 percent of people belonging to age group between 1-45 died by moving traffic.

2. In 2009, more than 2.3 million people were treated in emergency departments due to injuries caused by motor vehicle crashes.

3. In 2005, 70 billion of dollars was the added cost associated to deaths and injuries caused by crashes during transportation.

These incidents are not only responsible for the death tolls and injuries but they also produce added burden on the economy, mental traumas, added pain and high insurance premiums which is hard to afford for people. That is the reason why it becomes a major concern to address. To improve the situation transportation departments use a lot of funds for people’s awareness about use of safety belts and campaigns about obeying the traffic rules. Survey’s have proved that things improve with such campaigns a lot but there are still some loop holes. We can make the system much safer for people if these loop holes are acknowledged in proper manner. This is the place where technology comes into picture.

We can make transportation safer with the use of technology. Technology makes things more efficient and reliable. It helps us in not only studying the artifacts of crashes or designing the highways but also provides a way deeper insight into the future.

Intelligent Transportation System deals with storage and retrieval of huge amount of
traffic information efficiently and also analyzing it for the prediction of performance of the existing system. Analysis of such data helps not only in making transportation system more reliable but it also helps in setting future guidelines which in turn would reduce the traffic fatalities and financial burden created by them. Moreover, resources utilization can be optimized with help of transportation data analysis.

Regional Transportation Center (RTC)-Freeway & Arterial System of Transportation (FAST) and Transportation Research Center at UNLV have collaborated to address all such issues on the freeway I-15. Goals of the project are as follows:-

1. Conduct a comprehensive evaluation study on I-15 North Design Build Project

2. Analyze project implementation with respect to construction zone rules by which contractor (NCC) had to abide

3. Analyze data (quantitative and qualitative) collected from FAST for the various detour strategies

4. Study public outreach methods and their effectiveness

5. Conduct simulations of the network as well as major intersections

6. Develop guidelines for future major urban freeway detours.

We have provided a framework to analyze the qualitative data automatically. Qualitative data can be collected from various sources including news articles, messages generated by transportation department, user comments or complaints being
sent. We extracted the data from the famous social media website-Twitter, which is most popular among people to discuss various issues. The main problem is to first categorize the data and identify its type. Their type can vary a lot. For example, a complaint may be related to a signal timing or about an incorrect sign board. Moreover, emotions captured in every comment or statement is also of vital importance. It provides information about people’s satisfaction level with the existing system. Thus performance analysis can be done directly on this basis.

In this work, we propose a methodology which detects the exact sense in which each content word is used in text data and assigns a context to it and thus helps us identifying the type of data. This work is not only limited to qualitative data related to transportation. The concept can be widely applied in other text analysis applications as well.

1.3 Problem Statement

Seen from a broader perspective safety measures and rules and regulation formation is of major interest in transportation research. But formation of such rules depends a lot on the trends of present and past traffic data which includes not only speed-flow-density data but also user complaints and crash data which are generally in natural language. Automation of natural language processing is required to analyze this huge amount of data and to reduce human intervention which can be quite biased at times. Here we present a high level system architecture to analyze performance of transportation system.
Figure 1.1 shows the overall system architecture which is the final goal of the project. Qualitative and quantitative data analysis leads to the overall performance measure of the transportation system. A lot of techniques have already been developed to analyze text semantically but the main problem with state-of-the-art techniques is the huge complexity, requirement of long processing time and need of huge corpus. We have proposed a model which deals with such issues and provides respectable results. Text analysis field being quite broad, there are following areas which need to be addressed for the project from which word sense disambiguation, context recognition and sentiment analysis are the focus of the thesis:-

1. Semantic Analysis of Text

2. Word Sense Disambiguation

3. Automation of Context Recognition

4. Sentiment Analysis
1.3.1 Objective

Objective of this work is to present a novel and unique approach for context determination by word sense disambiguation for short texts (twitter data), providing the proof of concept and applying it on the data related to transportation. Moreover, we have applied sentiment scoring to measure people’s satisfaction for the existing transportation system around the globe.

1.3.2 Limitations of existing Text Analysis Systems

As mentioned earlier, text analysis field is no longer a new field. A lot of research has already been done and there are some commercial softwares available as well like Leximancer, WordStat etc. which are widely used for relationship extraction between
words and their visualization. Existing methods have several issues some of them are addressed below:-

1. Most of the methods work well only for long text documents but when it comes to short texts like user comments or tweets, systems do not perform very well due to lack of presence of the informative/content words.

2. Text is processed in very high dimensional space which generally have larger complexity and not adaptable to some applications.

3. These methods deal with both time and space complexities as they generally require huge corpus data for processing.

1.3.3 Solution

We have tried to address all these issue by the lexical semantic dictionary called WordNet[15] which maintains words in a semantically related hierarchy. This network allows us to define a metric such that the distance between two words can be defined in a semantic manner. These metrics help us in searching for the strong relationship between words in a statement and further allow us to select the word sense based on the strongest relations. Our idea is to exploit the property that more related words tend to appear together in a sentence or text. For example, He sat on river bank is a statement in which word bank clearly has the sense of a river bank but in statement like He deposits money in the bank has the sense of a finance company. So the words around the focus word define it’s sense.
Chapter 2 describes the problem statement in detail. The problem has been further divided into sub-problems. Literature survey for the sub-problems and limitations to those methods have also been discussed. Solutions to these sub-problems and results have been shown in next few chapters.
CHAPTER 2

LITERATURE SURVEY

2.1 Summary

This chapter discusses the advantages and limitations of current techniques that are available for concept discovery in text. We have also described the problem statement with an example and further categorized it into sub-problems. Motive of our work is to analyze the raw data to measure performance of the existing transportation system. Figure 2.1 shows the road map for this task:

Figure 2.1: Qualitative Analysis
2.2 Introduction

Let us start with an example of following sentences:-

1. They went fishing for some sea bass.

2. The bass line of this song is not strong.

By just looking at these two sentences we know that the first sentence has the context of fishing and in second we are talking about music or a song. Now the issue is, how to make a machine understand the texts and determine the contexts. There can be three possible ways to perform this:

1. Supervised Method: In this method we provide memory(labeled or classified data) to machine which helps it make decision based on this memory. It is quite similar to the way a human takes decisions as our memory or prior experiences help us.

2. Unsupervised Method: In this method we provide machine different scenarios from which it generates its memory(with some predefined logics) on its own and take decisions based on this memory.

3. Semantic Dictionary Method: This is quite a unique way in which machine takes help of a dictionary in which words or concepts are arranged by semantic relation not alphabetically. So the machine does not requires to have or build memory which actually is very efficient in terms of time and space complexity of prior learning methods but at the cost of some accuracy as semantic dictionaries are still new for such classification.
We have given detailed explanation to the first two techniques in the next sections and discussed the third one in more detail in the next chapter. Before getting into further details let’s start with some history of classification. Later, we switch to modern techniques of multi-class classification which will further lead us guide towards label-independent (general) classification.

2.3 History of Classification

Here we present a brief history of traditional classification systems. First we describe the types of existing algorithms and traditional methodologies then we describe the methods to represent the text in form of features/properties. Later we specify the types of classification problems that exist.

2.3.1 Categorization of Classification Based on Input Space

There are a lot of challenges to make robust text classification system. As we know language exists in several equivalent forms. A single statement can be said in unlimited number of ways. So the first challenge is to accommodate this large number of input space. Based on the input space methodology is categorized. There are three types of machine learning algorithms for classification:-

1. Supervised
2. Semi-Supervised
3. Unsupervised
Supervised and Semi-Supervised types of machine learning algorithms require training/labeled data from which we can train a classifier for decision making on test data. Accuracy of the classifiers decision making power depends on amount and diversity of the training data. Input space being too large it is a challenge to obtain this huge amount of training/labeled data. Unsupervised types of algorithms do not require training data. Unfortunately, this is a new area as compared to other two areas and a lot of research is yet to be done to make such algorithms efficient.

Third challenge is to remove noise from text data as most of the natural language documents have spelling mistakes and grammatical errors. These training documents lead us to erroneous models. Besides these challenges, complex learning task and computational efficiency of the learning algorithms are other challenges.

2.3.2 Traditional Classification

Figure 2.2 shows the traditional classification system’s flow of analysis.

![Figure 2.2: Traditional Classification Flow](image-url)
In the following sections we start describing the representation of text and features extraction. Further, we provide the definition of learning and describe each of the blocks shown in the flowchart.

2.3.3 Representation of Text

To make the learning algorithm efficient, text representation is very crucial. Every classification rule has some implicit assumptions and the representation of text should fit into those assumptions. Some fundamental representation forms are described here-

1. **Sub-Word level** In this kind of representation, text is represented by decomposing the words and their morphology. N-Grams are the examples of this decomposition. For example, in 2-gram( “bigram”) representation word “tool” will be represented by “t”, “to”, “oo”, “ol”, “l”. This approach is helpful in modeling the similarity of the words for example, word “computer” and “computers” are different with same meaning but most of their bigrams or trigrams are same. So duplicity can be avoided but at the same time we must also take care of the problem associated to it which arises because of different words with different meanings but almost same spellings. For example, we can consider the words “commuter” and “computer”. Although these words share most of their trigrams and bigrams still the information associated to them is totally different.

2. **Word Level**- This is so far the best proven representation because words are the basic meaningful unit of language. Although sometimes a single word can have dif-
different meanings in different context still they have little impact on the representation of a document or text. The main advantage of this decomposition is its simplicity as any piece of text can be easily tokenized in words. To avoid the ambiguity and maintaining the features length, only frequency of the word is recorded and the structure of document is ignored. Classifier generates a statistical model from these frequency and word tuples and makes decision based on this model.

3. **Multi-Word Level**- This representation is based on phrases and syntactic information in a text. These days we have some improved tools which can analyze large text with respect to their syntactical structure. In this scheme, we tokenize the text chunks of multiple words which incorporate syntactic information.

4. **Semantic Level**- This representation is not much evolved till now. Classifiers can have an optimal classification technique when it can understand the semantics of any text. Unfortunately, automatic extraction of semantics from a free text is not yet possible. With the advancement of technology researchers have come up with an entirely new way in which they have built lexical semantic dictionaries in which words are not organized in alphabetical order, rather a network of words is maintained in which semantically close words are kept together. We have explored this technique for our problem.

### 2.3.4 Features Selection

Basic features of any given document or text are its words. But this gives very high dimensionality to these features. So we further look to somehow trim the length
of these features.

**Trimming Feature Length**

This is the next stage of text classification after representation. It is required to filter out most important aspects of the text which helps us in reducing the size of processing data. It removes irrelevant attributes from the representation and thus improves computational efficiency.

One step for removing irrelevant features from text is called stop-word elimination. This elimination technique assumes that words like “the”, “a”, “and”, “for” etc. don’t have much relevant information. So, these kinds of words are removed from the representation. This algorithm basically removes the high frequency words.

The words which have very less frequency are also generally irrelevant. These words are also removed from the feature set by frequency thresholding technique. Besides these, some other techniques have also been proposed for further features reduction. One of them is called Mutual Information technique. This is one of the most common measure of relevance in machine learning algorithms.

We can further reduce the dimensionality of the feature set by stemming. Stemming is a technique which conducts a morphological analysis of words. It assumes that the words that are based on the same stem have same amount of information. So duplicacy in the feature set is removed. For example, words like “computer”, “computing”, “compute” and “computability” are projected on same attribute “comput”.


Feature Weighting And Normalization

TF-IDF(Term Frequency-Inverse Document Frequency) is the method which is generally used to weight the features in feature vector. The following formula is used to give weightage to the terms according to their importance. This “importance” is measured by the relevance of frequency of the words in the documents. A term which occurs more often in the documents must have more weightage. The term which occurs in more number of documents is less discriminating.

\[
TFIDF(t_k, d_i) = N_{(t_k,d_i)} \times \log \left( \frac{|T_r|}{N_{T_r(t_k)}} \right)
\]

1. In the above equation term \( N_{(t_k,d_i)} \) is the frequency of the word/term \( t_k \) occurs in text document \( d_i \).

2. \(|T_r| = \) total number of training documents.

3. \( N_{T_r(t_k)} = \) Number of documents in which term \( t_r \) occurs.

In the fields like information retrieval and text mining every feature in the feature vector is considered as a different dimension and every dimension has the measure or value related to the frequency of that feature(word frequency in our case). Cosine normalization is a way which gives a useful measure of how similar two documents are likely to be in terms of their subject matter. It is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them.
We normalize TF-IDF by using cosine normalization methodology. The following formula is used for normalization-

$$w_{ik} = \frac{TFIDF(t_k, d_i)}{\sqrt{\sum_{s=1}^{T} (TFIDF(t_s, d_i))^2}}$$

$w_{ik} = \text{normalization weight of word } t_k \text{ in text } d_i.$

$|T| = \text{total length of feature vector of document } d_i.$

### 2.3.5 Definition of Learning

A machine learning task is basically the task when a learner $L$ is given a training set $S$ of $n$ tuples

$$S = (\vec{x}_1, y_1), (\vec{x}_2, y_2), (\vec{x}_3, y_3), \ldots, (\vec{x}_n, y_n)$$

These tuples help learner to build a statistical model. This process of generating statistical models is called learning. The sample space $S$ is supposed to be drawn independently and identically distributed with an unknown but fixed distribution $P(\vec{x}_n, y)$. Performance of any classification rule $h$ is measured by the parameter called Risk and the performance of this risk is measured by Loss function $L(h(\vec{x}_n), y) \epsilon R$. This loss function is an appropriately chosen measure of error. It measures how far
the class label is predicted from the observed class label. Risk $R(h)$ factor is further defined as

$$R(h) = \int L(h(\vec{x}), y) dP(\vec{x}, y)$$

Role of learners is to try and find a classification rule $h$ by minimizing this risk factor. As we have stated, probability distribution of $S$ is unknown so finding such classification rule($h$) is not directly possible. Different learners try to find the distribution with the help of the training set $S$. This is the reason why learners performance is based on size of the training set $S$ as the accuracy of probability distribution approximation increases with larger size of $S$.

![Figure 2.3: Machine Learning](image-url)
2.3.6 Types of Classification Problems

These are the settings which have evolved over time.

1. **Binary Setting**- In this setting classifier answers only in yes or no that is whether a document belongs to one category or not. This setting arises when each input text feature vector can have only 2 labels (0,1 or 1,-1).

\[
S = (\vec{x}_1, 0), (\vec{x}_2, 1), (\vec{x}_3, 0) \ldots \ldots (\vec{x}_n, 1)
\]

2. **Multi-Class Setting**- Some classification tasks have more than two classes scenario. For example, a customer complaint at a service hotline might require to be redirected to one out of ten customer representatives. This means that classifier has to make decision over ten choices/labels.

\[
S = (\vec{x}_1, y_1), (\vec{x}_2, y_2), (\vec{x}_3, y_3) \ldots \ldots (\vec{x}_n, y_n)
\]

3. **Multi-Label Setting**- This situation arises when there is no one to one relationship between input vector and labels that is the input can have multiple labels.

\[
S = (\vec{x}_1, S_1), (\vec{x}_2, S_2), (\vec{x}_3, S_3) \ldots \ldots (\vec{x}_n, S_n)
\]
where

\[ S_1, S_2, S_3 \ldots S_n \subseteq Y \]

\[ Y = \{ y_1, y_2, y_3 \ldots y_n \} \]

2.4 Learning Classifiers

There are a lot of machine learning algorithms which are used these days in text classification. Some of them have been described here.

2.4.1 Naive Bayesian Classifier

Naive Bayesian Classifier\[1\] is one of the simplest classifiers which is based on Bayes probability rule. Naive Bayesian Classifier uses the assumption of independence between the affect of an attribute value on given class from the values of other attributes. For example, a ball may be considered as tennis ball if it has green color, round in shape and has around 3-4 inches of diameter. Although, these three parameters may have some dependencies over each other but Naive Bayesian Classifier assumes that these parameters are independent.

Working of Naive Bayesian Classifier
If we have a labeled training set $T$, with labels $y_1, y_2, \ldots, y_k$. If we have $n$ feature value $x_1, x_2, \ldots, x_n$, then each sample can be represented by a $n$-dimensional vector $\vec{X} = \{x_1, x_2, x_3, \ldots, x_n\}$. The classifier develops an statistical model based on this input data.

For any test vector $\vec{Z}$, classifier classifies it to the class which has the highest posteriori probability which is conditioned on vector $\vec{Z}$. Mathematically, class $y_i$ is labeled on $\vec{Z}$ if

\[
P(y_i \mid Z) > P(y_j \mid Z) \forall i \neq j
\]

By Bayes’ Theorem

\[
P(y_i \mid Z) = \frac{P(Z \mid y_i) P(y_i)}{P(Z)}
\]

Now, to maximize $P(y_i \mid Z)$ we have to maximize the right side. Classes are considered to be independent and equally probable so all $P(y_i)$ are equal and $P(Z)$ is also same for all. So, the classifier tries to find the maximum $P(Z \mid y_i)$. To find $P(Z \mid y_i)$, classifier assumes the independence of class conditional probabilities. So, mathematically,
\[ P(Z \mid y_i) \approx \prod_{j=1}^{n} P(x_j \mid y_i) \]

where

\[ Z = \{x_1, x_2, \ldots, x_n\} \]

To compute \( P(x_j \mid y_i) \), classifier first calculates the mean and variance of \( Z = \{x_1, x_2, \ldots, x_n\} \) and then assumes \( Z \) to have gaussian distribution with calculated mean and variance corresponding to a particular class, with pdf:

\[ p(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \]

### 2.4.2 Rocchio Algorithm

This algorithm was proposed by Rocchio in 1971 for Smart Information Retrieval Systems\[19\]. This was based on refining user query terms using the feedbacks indicating the result as irrelevant or relevant. This is the basic formula that was provided by Rocchio in 1971.
\[
Q_{\text{new}} = Q_{\text{old}} + \left( \frac{1}{|R_{\text{docs}}|} \right) \ast \sum_{d_i \in R_{\text{docs}}} w_{ik} - \frac{1}{|\bar{R}_{\text{docs}}|} \ast \sum_{d_i \in \bar{R}_{\text{docs}}} w_{ik}
\]

where,

\[ Q_{\text{old}} = \text{old query vector} \]
\[ Q_{\text{new}} = \text{resultant query vector} \]
\[ R_{\text{docs}} = \text{relevant retrieved set based on } Q_{\text{old}} \]
\[ \bar{R}_{\text{docs}} = \text{irrelevant retrieved set based on } Q_{\text{old}} \]
\[ w_{ik} = \text{document feature vector} \]

For text categorization this algorithm has been made efficient by Ittner in 1995\cite{20}. Ittner assigns some weightage to all the terms in right hand side of the equation. The first term which are old queries were omitted in this new form. This model builds a prototype for each class based on given training dataset. The prototype is expressed as a vector \( \bar{c}_i = \langle w_{i1}, w_{i2}, ..., w_{iT} \rangle \) So the new equation becomes:-

\[
w'_{ck} = \beta \ast \left( \frac{1}{|R_{c}|} \right) \ast \sum_{d_i \in R_{c}} w_{ik} - \gamma \ast \left( \frac{1}{|\bar{R}_{c}|} \right) \ast \sum_{d_i \in \bar{R}_{c}} w_{ik}
\]

where,

\[ w_{ck} = w'_{ck} \text{ for } w'_{ck} > 0 \]
\[ 0 \text{ otherwise} \]
2.4.3  K-Nearest Neighbour Algorithm

This is one of the simplest learning algorithm used for classification. It uses the
closeness of features in training samples to categorize a document. The class which
is most common amongst the k nearest neighbours of the document is assigned to it.
In this algorithm training samples are saved as a multidimensional feature vector and
a label defined to it. K is a user defined constant to restrict the analysis for k number
of neighbours. The distance metric which is used to find the k nearest neighbours is
generally Euclidean Distance. But in case of text categorization this metric cannot be
used. Hemming Distance is used for such problems. Improvement in metric selection
definitely improves the performance of this algorithm.

The algorithm has major drawback when training data has a lot of samples labeled
with same class. This makes the feature domain skewed and there is high probability
that most of the classification done by the system will predict the same class.

These are simple algorithms which over the period of time have improved a lot. Im-
provement are done by tuning, weighting and superposition of other techniques. After
these basic methods an advance method was developed for the classification which is
known as Support Vector Machines (SVM) which has been described as follows:

2.4.4  Support Vector Machines (SVM)

Support Vector Machine method was proposed by V. Vapnik and C. Cortes in
1995[5]. It was further explored and improved on many occasions. [3, 4, 6] and [7]
describe such improvements in detail.

It is a supervised learning technique used to classify data in different categories. In this method training feature vectors are mapped to a space where vectors belonging to different categories can be separated as wide as possible. This method uses the concepts of linear algebra for the mapping of features in other dimension, which makes it a non-probabilistic learning model for classification.

In binary setting this method creates a hyperplane in the higher or infinite dimension such that summation of the distance of nearest training samples of different categories can be maximized. In case of multi-class setting, this creates a set of such hyperplanes. Test data feature vector is also mapped to the same higher or infinite dimensional space and depending on the gap in which this vector lies category is assigned of that side. Figure 2.4 makes thing clear.

\[ \text{Figure 2.4: Support Vector Machine} \]
SVM was originally developed as a linear classification technique which means that classification was based on linear combination of characteristics. Later on non-linear SVM techniques were developed in which a trick called Kernel Trick was used to maximum margin hyperplanes such that it gets fitted in the transformed feature space.

So, these are the basic classical classification techniques that have been used over and over till now. A lot of improvements have been done on these techniques specially by combining two or more of these techniques. Moreover, SVM opens up huge space for improvements in non-linear classification. Development and application of different kernel functions have improved its performance a lot.

From all these techniques we also infer one thing that data organization, feature extraction and application of these techniques is a cumbersome and complicated task which require large and complex mathematical calculations.

The other problem with these methods are the lack of usage of semantic information in the text. This is a big one. Structure and Semantics are two important informative parts of text. All these methods can deal with structuring or grammar of the sentence during feature selection. But the use of semantics(meaning) in such algorithms are not very successful till now.

To make things easier for development researchers started to build lexical semantic network of words in a language so that traversals can be done through the network based on the meanings of words and moreover, this also introduce the concept of
distance metric by meanings of words. There are several lexical semantic dictionaries which exist today. Although each such network have few limitation till now but this surely will be the future of categorization and learning. Such networks are also quite useful in disambiguating the word senses that means we can categorize text data by disambiguation of senses of words that are present in the text.

2.4.5 Categorization Based on Content Similarity

Although its quite easy to read and sense the similarity between two documents but sensing similarity by machine is a tedious task. Traditional content similarity measure algorithms are mostly centered around shared word processing. But it simply fails when there are no shared words between two contents. Therefore those algorithms generally works for long texts but fails when it comes to shorter ones. Moreover natural language flexibility allows human to form much different sentences using same words. So documents having a lot of sharing words can differ a lot. Some other methods requires a lot manual preprocessing which again is unacceptable with the amount of data present today. Most of the existing similarity measure analyzer can be broadly classified in the following three categories:-

1. Bag of Words Method
2. Corpus-Based Method-LSA [21]
3. Descriptive Method
Bag of Words Method

These methods are commonly used in information retrieval (from web) systems. This methodology requires a list of all meaningful (in the predefined context) words in the language which could go to million (say m). Every document is then represented by text data vector in this m-dimensional space. Queries are again represented in this m-dimensional space. System fetches out the information based on the number of words between the vectors that are similar.

Limitations

This methodology has a lot of issues of space complexity. Besides, documents having shared words may not be close to each other as well as it is also possible that similar documents may not have a lot of shared words. Other than that this method only works for long texts. Due to high dimensional complexity this method omits the function words the, to, for etc. but such words have major importance in relation measures and carry major structural information.

Corpus-Based Methods

This is one of the recent development in similarity measurement. This method uses statistical distribution of words in huge dataset to measure the similarity. Latent Semantic Analysis[21] is one of such methodologies. These methods use context based corpus. A word by context matrix is developed which is further decomposed using singular value decomposition. Then threshold based truncation is performed on the diagonal matrix produced. This step reduces the dimension of word by context matrix.
But again this method has its own limitations.

**Limitations**

These limitations are mainly because of incompleteness of corpus. If the input sentence is from unconstrained domain that is it does not have its context defined, then this method won’t work. Furthermore, SVD has its own computational limits. Like **Bag of Words** method, LSA also ignores the syntactic information present in the sentence.

**Descriptive Features Methods**

The third category for similarity measurement is descriptive features methods. In this method, every statement in the text is described by a set of predefined features semantically. For example, a noun word can be a HUMAN(yes or no), Organization(yes or no), Place(yes or no) or something else. Features are prepared from two categories **Primitive** or **Composite**. Primitive features are those which we get from comparing every single word in a sentence to that in other sentence. While composite features are generated by comparing combination of words in two sentences. A final feature vector is made by combining these primitive and composite features. Similarity measure is further obtained by passing these features to pre-trained machine learning classifier.

**Limitations**

Main limitation of this method is the definition and extraction of features from a
text. Concrete concepts may have a set of well defined features but this is not valid for abstract concepts. Moreover, training a classifier requires training data set which could be impractical or quite tedious to develop.

2.4.6 General Context Recognition- Short Text

Next important aspect for text analysis is context recognition which means extraction of the sense in which text is about. This problem is quite related to text categorization problem in which categories are unknown and selected category or sense describes one word for that document or sentence. A lot of research have been done in this but those techniques generally categorize the documents in some predefined categories and most of them requires large corpus for training. We have specifically focused on techniques which use WordNet which is lexical knowledge based semantic network of words. The basic problem in context recognition is to disambiguate meaning of words. For eg. word Bank is used for financial companies which hold money and it is also used for River Banks. This problem is known as Word Sense Disambiguation. Few of them are as follows:-

Conceptual Density

This method was proposed by Agirre and Rigau[16]. This method deals with the relatedness among words and word senses. The term conceptual density is used to capture the strength of closeness of a set of concepts in the hierarchy like WordNet[15].
This utilizes the relation between weights given to word senses and the size of the sub-hierarchy which contains all the word senses. So, a target word in a given text is disambiguated using the surrounding nouns that is the hierarchy which has the highest conceptual depth is chosen. Highest conceptual depth is actually the smallest hierarchy which has most context words.

**Unsupervised Context Discovery**

Enrique and Suresh\[23\] also provided an unsupervised approach to discover context automatically which combines different NLP fields like, named entity recognition, knowledge acquisition and word sense disambiguation. They refer this technique as general named entity recognition. The problem with this technique was that it is domain dependent and requires prior knowledge or corpus of the domain of interest.

**Decision List**

This method is completely based on prior knowledge or corpus. It requires a training set in which the target word is tagged with the sense in which it is used. Then frequencies are calculated for, word-sense tuples, bigrams and trigrams or each such tuples and also the surrounding words. Performance of this method depends a lot on the prior data. With a huge tagged corpus this works well. Issues with this technique is to arrange such sense-tagged corpus which requires a lot of human effort.
2.5 Conclusion

As we see from the literature survey there are a lot of issues with the existing text categorization systems. In order rectify them and for ease of implementation we have proposed a hypothesis which would help us in general categorization of short text by disambiguating word senses. In next chapter, we have given a brief introduction of WordNet which is one of the most powerful lexical semantic dictionary today.
CHAPTER 3

WORDNET AND METHODS

3.1 Summary

In this chapter we have described the text categorization and word sense disambiguation methods which take the help of knowledge based semantic dictionary known as WordNet. We start with brief introduction of WordNet and the description of the semantic network. We further describe the methods that have already been proposed using WordNet.

3.2 WordNet

Considering the example described in chapter 2 again, which deals with the following sentences:

1. They went fishing for some sea bass.

2. The bass line of this song is not strong.

Labeling such small statements with a context is a cumbersome task. The problem being the presence of less amount of informative words as well as it is extremely difficult to have training corpus for all such sentences. This is where knowledge based semantic dictionaries come into picture. Semantic dictionary which we have used
in our work is call WordNet. WordNet is one the most advance lexical database for English language.

Our framework is based on WordNet and it is quite efficient for dealing with such small sentences. It is also easily applicable to long texts which can further be tokenized in sentences. It is a breakthrough in the field of General Context Recognition without using any corpus.

WordNet was created under the direction of Professor George A. Miller at the Cognitive Science laboratory in Princeton University. It is lexical database which basically groups English language words in the sets of synonyms called synsets. A word can be in multiple synsets depending on the variation of meanings it has. It also keeps track of the various relationships which these synsets have semantically.

For example, some of synsets of word **book** are the following:-

1. **Synset('book.n.01')** - a written work or composition that has been published (printed on pages bound together).

2. **Synset('book.n.06')** - a collection of playing cards satisfying the rules of a card game.

3. **Synset('book.n.07')** - a collection of rules or prescribed standards on the basis of which decisions are made.

4. **Synset('book.v.01')** - engage for a performance.

Now, these are just four of the fifteen synsets the word **book** has. It can be easily figured out that the letter **n** in the synset refers to **Noun** and the letter **v** refers to
Verb. So, it distinguishes between noun, verb, adjective and adverbs. Moreover, the dictionary provides the short definition or sense in which the word is kept in that synset.

All these synsets are arranged in a network such that they follow an IS-A hierarchy (similar to the hierarchy which is followed in object oriented programming language like JAVA). For example, everything is an object and object can further be classified into abstract and physical object. So it is a tree of concepts. A dog is an animal and cat is also an animal but cat is not a dog. So, cat and dog do not follow IS-A hierarchy but they follow the same with animal.

This kind of network is quite useful in estimating a semantic distance between two words. Question to think:- does it really help in estimating the distance between meanings? The answer to this question is yes because, it helps in estimating the distance between synsets not the words. So, now the main problem is to correctly
find the most relevant synset for a word in a given sentence.

Besides this IS-A hierarchy, WordNet also maintains a number of other semantic relations. Some of which are described here:-

1. **Hypernyms** - All parent synsets are called hypernyms of child synsets in IS-A hierarchy.

2. **Hyponyms** - All child synsets are called hyponyms of parent synsets in IS-A hierarchy.

3. **Holonyms** - A synset A is a holonym of synset B if B is a part of A (Tree is a holonym to branch).

4. **Meronym** - A synset A is a meronym of synset B if A is a part of B (Branch is a meronym to tree).

Such information makes WordNet quite powerful but still it has a lot of limitations. The first and foremost being the non-relatedness between different POS. WordNet does not have very strong relations between different parts of speeches. For example:- Adverb **costly** and Noun **cost** do not have any network bond that is we cannot find similarity value between these these words, although they seem quite similar to each other.

The other problem with WordNet is it incompleteness. Incompleteness in the sense of etymological information which most of the dictionaries poses. It also does not include the information about pronunciation and forms of irregular verbs.
3.3 Literature Survey - Text Categorization Using WordNet

In this section we have explained the methodologies being used in past to categorize documents using WordNet.

3.3.1 Rodriguez Method

It started with the study done by Rodriguez 97 where they enhanced neural network learning algorithms for significant improvement in classification accuracy. They used WordNet based information as an additional source of information to improve the existing system. The information was used in the training phase as features and machine learning algorithms which were tested were Rochhio and Widrow-Hoff. For this purpose they used synsets provided by WordNet. Word sense disambiguation was done manually in this method which required quite lot of human effort which made the system non-automatic.

Results of this experiment showed exceptional improvement in the classification by these learning algorithms. Moreover, it was proved that the classification of low frequency categories can also be effectively improved through this technique. This was a major breakthrough. It attracted researchers to concentrate in this field.

Limitation:- The system was not automatic as it required human effort for word sense disambiguation. Besides there method took advantage of the corpus on which it was tested. Actually they used corpus headers also as information source which actually were quite informative regarding the topic of the document or class. More-
over, this method used WordNet to basically improve the performance of algorithms used in neural networks, using the same bag-of-words representation of text which means they did not use the full power of WordNet which could be helpful even in the representation of text.

### 3.3.2 Classification Using WordNet Hypernym

This method was proposed by Scott and Matwin[11]. They explored the problems of manual word sense disambiguation with the existing Rodriguez method and its other limitations. They tried to address it by using not only synonyms in WordNet network but also utilizing power of hypernyms.

To avoid the advantage of corpus headers, they chose three corpus for their study.

1. **Reuter-21578**

2. **USENET**

3. **Digital Tradition(DigiTrad)**

   Reuters and USENET corpuses have header advantage but DigiTrad is the collection of folk songs lyrics and it so the assigned label would be more related to semantics rather than words itself. They performed some experiments for proving how the introduction of corpus like DigiTrad makes thing difficult in categorization.

   This algorithm had three steps:-

   1. Part of Speech Tagging based using Brill tagger.
2. Global list of all synonyms and hypernyms of each noun and verb is maintained.

3. For all synsets in the global list a density value is estimated which is number of occurrence of synset divided by the total number of words in the document.

To make the system semantically strong, they influenced the calculations of Step 3 by varying number of synonyms and hypernyms in the global list. This was done on the basis of a controlled parameter \( h \) which was called the **height of generalization**. They claimed that setting the parameter correctly and large enough helps disambiguating word senses automatically as synonymous and hyponymous words map to common synsets which in turn increase their weightage in training. **Limitations:**

Although this method answered many issues in the existing Rodriguez method but still it seems a bit inefficient. First of all, It uses Reuters and USENET which again influence the training by their headers. Moreover, use of WordNet seems a bit inefficient too which means the method of word sense disambiguation is not very convincing. Secondly, global list is maintained using verbs and nouns only but sometimes adjectives and adverbs are also extremely informative. Besides, YES this method requires corpus, which is again a turn off for small sentences/tweets/updates.

### 3.3.3 Using Hood Algorithm

In Hood construction[13], the advantage of considering each hierarchy in WordNet as a category is used. Hood is intended to define an appropriate middle level category.
For a synset $s$, hood is defined by considering the hyponymy synsets as vertices and
and the edges joining them as direction edges of a graph. Then, largest connected
subgraph containing $s$ is defined as the hood of $s$. This subgraph contains only the
descendants of an ancestor of $s$ and contains no synset that has a descendant that
includes another instance of a member of $s$ as a member. Representation of hood is
done by the root of the hood.

For word sense disambiguation hoods are constructed for each synset in WordNet
hierarchy. These hoods are used to select the sense of ambiguous words in a text
document. Then a two step process is followed for disambiguating the words. Both
steps have a common process called **Marking** which is used just to visit synsets and
maintain a counter for each synset. First step is similar to a training step in which the
**Marking** method is called over a collection of documents. For each word $w$ **Mark-
ing** finds all the instances of $w$ in the WordNet and for each instance parent-child
hierarchy is followed up to the root while incrementing the counter of each synset it
visits. In the first step, **Marking** is called for each occurrence of word $w$ in the
whole collection of documents and a counter is maintained to store the number of
times **Marking** method is called for a word $w$. So, in the first step, a global count is
produced for each synset. In second step, **Marking** method is called for each word $w$
in a single document and the counter is maintained in the similar fashion as in step 1.
This produces the local count for each synset within a document. Then, a difference
parameter is estimated which is given by the following equation:-
\[
\text{difference} = \frac{LocalVisits}{LocalCalls} - \frac{GlobalVisits}{GlobalCalls}
\]

For each sense of w, difference is calculated on the root of the hood. Sense having no hood or local count less than 2 are set to have difference equal to zero and for senses having multiple hoods difference is set to be the maximum values over all hoods. The sense having the largest positive value of difference is selected as the sense for the word w.

**Limitations** There are two limitations with this algorithm:-

1. Lot of calculation is required for the preprocessing.
2. Big collection of documents is required for the training purpose.

### 3.4 Conclusion

So here we provided the methods which use WordNet for word sense disambiguation and text categorization. In next chapter, we have described our hypothesis in detail.
CHAPTER 4

PROPOSED METHOD

4.1 Summary

In this chapter we have described the method which we have proposed for Word Sense Disambiguation and concept discovery through it. We start with an example and continue through the content similarity measurements provided by WordNet. These measurements provide different metrics which can be used to give a number to the closeness of two synsets. It helps us in finding the strongest relationships within different synsets which is the soul of our method.

4.2 Proposed Hypothesis

We propose a completely unsupervised hypothesis for context recognition for short text(sentence) by word sense disambiguation which doesn’t require any prior corpus knowledge. Our main idea is that in general statements the informative words have strong relatedness. For example: In sentence, \textbf{He is sitting on the river bank}, the information word river will be closely related to the word bank which has \texttt{river\_bank} sense not a \texttt{financial\_company} sense. Wordnet which is the lexical knowledge based semantic network comes out as a savior in this task. It has ISA(is-a) hierarchy of
Nouns (which actually are concepts) and hierarchy for Verbs as well. We utilize power of this semantic relation in our task. Based on the recognition of context for short text we have also proposed the context recognition for documents, in which we parse the document in short texts or sentences, determine the context of each of this chunk and then look for the strongest relationship between these contexts. Here is the proposed architecture from a high level:

![Figure 4.1: Context Determination Model for Documents](image)

This architecture is totally based on Wordnet Network, which itself has some issues. For example, Verbs hierarchy is not very well formed in Wordnet. Wordnet also does not have any semantic hierarchy for Adjectives, Adverbs or other parts of speech. So, it becomes difficult for context recognition of sentences like *This thing is costly*, where actual context is an adverb. So, to eliminate such issue we have used a technique which estimates **Derivationally Related Forms** of such POS. So, for example, word *costly* has the derivationally related form *Cost* or *Price*.

So, we convert all words which are not Nouns into it’s closest derivationally related
Noun and then estimate the closeness between those Nouns or contexts. Here is the proposed lower level model which is actually the Process used in document context recognition model.

So, as we know, a noun word can have multiple senses and several words can be used for a particular which are nothing but the synonyms of the word in that sense. Wordnet provides a hierarchy of synonyms of concepts which are called *synsets*. For example: the word *bank* in Wordnet as 18 different synsets which means this is used in 18 different senses out of which 10 are in noun hierarchy and rest 8 are in verb

![Context Determination Model for a Sentence](image-url)

Figure 4.2: Context Determination Model for a Sentence
sense. So, for a sentence like, **He is sitting on the river bank** we have to select one sense out of these 18 senses which corresponds to **river_bank** sense.

For this purpose we have to find the closeness defined by a Metric on these words. There are many existing similarity measure metrics available. Some of them are :-

### 4.2.1 Path Similarity

This scheme is provided measure of similarity related to path[23], [24]. It is based on the shortest path between two synsets in IS-A hierarchy of WordNet. Words that are more related are expected to be near each other. For example, words **boy** and **girl** are the child of node **human** are more related than to words **boy** and **dog** who share the common node **animal** which is the parent of **human**. Therefore, distance between words and closeness is inversely proportional.

### 4.2.2 LCH Similarity

LCH similarity[25] is also related to the shortest path between two synsets or concepts. This shortest path is further scaled by maximum path length in the IS-A hierarchy in which they occur.

### 4.2.3 WUP Similarity

**Wu and Palmer, 1984[26]** proposed this measure which not only takes the idea of shortest path but also the depth of their least common subsumer (LCS), which is nothing but the depth of the first node that is shared by two concepts climbing up
There are some other similarity measures as well like Resnik \cite{27} similarity, Lin \cite{29} similarity and Jiang and Conrath \cite{29} similarity. But all these measures are based on information content of concepts based on some corpus. Out of these measures, we chose WUP similarity which not only depends on the shortest path between two synsets but also their depth. Closeness between two synsets increases exponentially with depth in the hierarchy. Path similarity on the other hand uses only the shortest path to estimate the similarity. Reisnik similarity uses the information content of a synset from a corpus which we wish to avoid in our hypothesis.

4.3 Context Determination Algorithm - Short Text

Although it has already been discussed that most informative words determine the concept of the sentence. But WordNet itself has some limitations. Sentences are comprised of different part of speech and context can be associated not only with Nouns but also with other parts of speech(POS). For example, if we consider a general sentence This place is costly, the concept is determined by adverb costly but unfortunately WordNet does not have strong hierarchical trees for POS other than Nouns. So, we have to convert all POS to closely related Nouns. Adverb costly is related to Noun Price or Cost. This can be achieved by examining derivationally related forms of other POS.
Here we describe the steps which are followed:-

4.3.1 Step 1: Preprocessing

This is the preprocessing step. Each word of the sentence should be tagged with its POS. We used Python NLTK for this purpose. Then non informative function words like prepositions should be removed. So that we have a list of tuples having information words and their POS.

4.3.2 Step 2: Converting Words to Concepts

In this step, we convert all non-Noun information words to their respected derivationally related Noun synset. WordNet provides list of all lemmas with POS tags which are derivationally related to the input word. For convenience, we chose the first Noun lemma that occurs in the list.

Figure 4.3: Synsets Network
4.3.3 Step 3: Similarity Evaluation

Following Step 2, we have the list of all noun words and synsets of other derivationally related forms of different POS. Now, a synset list is made corresponding to each word. Then WUP similarity is calculated between synsets of different words. We made all combinations of synsets from different words and calculated WUP similarity for each combination. Then we find the summation of similarity measures between each synset in a combination. The combination which has the maximum value for this summation disambiguate the word sense and thus we find the exact concept for that sentence.

Figure 4.4: Similarity Measure
Moreover, we further tag the sentence with these synsets as it’s general context.

4.4 Conclusion

In the next chapter, we have presented the results using our hypothesis on short texts. Later we use the same for text classification.
CHAPTER 5

SENTIMENT EXTRACTION

5.1 Summary

Social networking sites like Twitter and Facebook are the most popular forums among people to discuss various issues these days. It is in particular of researchers interest to study people’s opinion and sentiments regarding the subject of their study. In this document we have discussed the methodology to analyze performance of current transportation system by analyzing sentiments of public tweets related to transportation and traffic.

Based on sentiment measure, a score was given to each tweet and a color map was created based on these scores. This colormap was tagged on google maps using google maps APIs to explore satisfaction or problems of existing transportation system around the globe.

5.2 Introduction

Sentiment analysis refers to use the natural language processing techniques to extract the subjective information in data. Performance of any existing product or system can be studied with the help of data analysis and user reviews. Evaluating
user comments manually can be a cumbersome task as millions of user comments and
tweets can be generated daily for a regular topic. There are a lot of techniques that
have been developed over the period for the automation of such task but unfortunately
when it comes to short texts these techniques are not very efficient.

In this chapter we have focused on extracting transportation related data from twit-
ter and measuring sentiments in it. We maintained a predefined list of words for
positive and negative sentiments. First we have discussed the method to extract
transportation data using twitter APIs.

5.3 Data Extraction

Twitter is one of the most popular social networking sites where people posts and
discuss about various issues and topics. Data posted by people is known as tweet.
Twitter provides users with privacy setting by which users can control the visibility
of their tweets. Tweets are publicly visible by default but user can restrict it to their
followers only. Users can group posts together by topic or type by use of hashtags
? words or phrases prefixed with a ”#” sign. Similarly, the ”” sign followed by a
username is used for mentioning or replying to other users. To repost a message from
another Twitter user, and share it with one’s own followers, the retweet function is
symbolized by ”RT” in the message. Twitter determines the trending topic among
people with the help these hashtags. Moreover, it also allows developer to extract the
data related to their fields of interest. Twitter asks third party developers to register
their application on their website before allowing them to extract any data. Before
2010, twitter used to ask developers to use their login and password for that. But due to privacy concerns they had to abandon this methodology and provide developers a safer way for that. Twitter applications are required to use OAuth, an authentication method that does not require users to enter their password into the authenticating application. It is an open standard for authorization purpose.

OAuth provides two authentication keys for the application which are known as access token and access token secret and two authorization keys for the developer known as consumer key and consumer secret. These keys are unreadable character strings which encodes user login information and application’s registration information. For extraction we developed a code in Python which creates an object of class StreamListener which is used to wait for live feeds(streaming data). Moreover, it captures and appends those feeds in .json file. Each live feed is is an structured object which not only contains the tweet but also it’s language, datetime stamp and other related information. We captured data only in English language.

Although twitter allows developer to extract and analyze data but it has forced few terms and conditions over it. Twitter does not allow developers to distribute the data. Only registered applications are allowed to stream and use the data. We used the following keywords to extract the transportation related data:-

1. traffic
2. transportation
3. freeway
4. highway

5. road

6. accident

5.4 Methodology

Extracted data was processed before applying the scoring technique. Following processing steps were applied on the data:

1. Non alphanumeric characters filtering.

2. Stopword (eg. punctuations) and other non informative words removal

3. Lower case conversion

4. Stemming: so that words like happiness and happy are considered as same.

To measure the sentiments in tweets, two lists of words corresponding to positive and negative sentiments were maintained. Scores were given to the tweets based on the number of positive or negative sentiment words present in it. For each positive word in the tweet the positive score was increased by the value 0.5 and similarly for each negative word the negative sentiment score was increased by 0.5. The scores were subtracted to find actual sentiment measure. Once the processing was done, we scaled all scores between 0 to 1 based on minimum and maximum value of sentiment measure. Here we present the flow chart for data extraction and analysis:-

Figure 5.1 Sentiment Analysis of Twitter Data
Based on the sentiment scores, a colormap was made to visualize sentiments. Moreover, tweets having the location information were tagged on google maps using the google maps APIs and the colormap. This analysis clearly pointed out the areas where people are suffering more due to inefficiency of the transportation system and in need of immediate attention.

5.5 Conclusion

We followed the algorithm described above and extracted tweets related to transportation having keywords(hashtags) traffic, transportation, accident etc. We removed the function words like prepositions from the tweets and used stemming(as described in previous chapters). In the experiment most of the tweets were classified
as having negative emotions which is understandable as most of the time people usually complain about traffic conditions. This is by far the unique and effective method to study the traffic performance from the people feedback directly.
CHAPTER 6

RESULTS & CONCLUSIONS

6.1 Summary

In this chapter, First we give the proof of concept for our novel approach of context recognition by describing an example in detail. We further describe the test setup which was developed to test our proposed hypothesis. Concepts and limitations of our algorithm and propose improvements for the future work. We have also provided the results of sentiment analysis on the data which clearly shows the places where there is need of immediate attention for transportation system improvement.

6.2 Proof of Concept for Context Discovery

With this example we prove our concept:-

Lets consider two close sentences:-

1. *Cricket is played with a bat*

2. *Cricket flies higher than a bat*

We can clearly see that in the above sentences bat and cricket have different meanings. In first sentence, cricket and bat are related to a sport, while in second sentence these are two animals. Following the algorithm mentioned for first sentence we see
the following relation which maximizes the summation of WUP similarities between synsets:

Figure 6.1: Similarity Measure 1

Applying the algorithm mentioned for the second sentence we see the following relation which maximizes the summation of WUP similarities between synsets:

Figure 6.2: Similarity Measure 2
Measure of closeness, for first sentence is maximized when the combination of shown synsets are considered which by definition correspond to the sport cricket which is played with a bat. Similarly for second sentence the measure of closeness is maximum with the synsets which correspond to flying of insect cricket and bat. Moreover, here we provide some more examples which clearly prove our concept.
Figure 6.3: Proof of Concept
6.3 Test Setup

Here we describe the test setup we developed for testing our algorithm. We chose five words at random having disambiguous senses depending the sentences. Words chosen for this purpose were:

1. Turn
2. Traffic
3. film
4. Draw
5. State

For each word in the list above, we randomly make a dataset of few sentences in which they have different meanings. We apply our algorithm for each sentence having that word and give a score (0 or 1) manually. If the definition of selected synset is relevant to the sentence we give a score 1 if not we give a score 0. We chose the most informative synset in the sentence to be the context.

We applied the algorithm over all sentences. Based on the score, we estimated the percentage of 1’s which is nothing but the percentage of accuracy of the algorithm.

Here are the example dataset developed:
<table>
<thead>
<tr>
<th>SENTENCE</th>
<th>SYNSET</th>
<th>DEFINITION OF SYNSET</th>
<th>POS</th>
<th>SCORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web links communication generate traffic for your site.</td>
<td>Synset['traffic.n.03']</td>
<td>the amount of activity over a communication system during a given period of time</td>
<td>Noun</td>
<td>1</td>
</tr>
<tr>
<td>Human traffic is the illegal trade of people.</td>
<td>Synset['traffic.n.02']</td>
<td>buying and selling especially illicit trade</td>
<td>Verb</td>
<td>0</td>
</tr>
<tr>
<td>Network choked because of huge data traffic.</td>
<td>Synset['traffic.n.01']</td>
<td>the aggregation of things (pedestrians or vehicles) coming and going in a particular Ad</td>
<td>Adj</td>
<td>1</td>
</tr>
<tr>
<td>Traffic in rush hours cause major delays.</td>
<td>Synset['traffic.n.03']</td>
<td>the amount of activity over a communication system during a given period of time</td>
<td>Ad</td>
<td>0</td>
</tr>
<tr>
<td>Travel time analysis is important to maintain traffic.</td>
<td>Synset['traffic.n.03']</td>
<td>the amount of activity over a communication system during a given period of time</td>
<td>Ad</td>
<td>0</td>
</tr>
<tr>
<td>Car accident caused huge traffic jam on the highway.</td>
<td>Synset['traffic.n.03']</td>
<td>the amount of activity over a communication system during a given period of time</td>
<td>Ad</td>
<td>0</td>
</tr>
<tr>
<td>Traffic is the major problem in big cities.</td>
<td>Synset['traffic.n.03']</td>
<td>the aggregation of things (pedestrians or vehicles) coming and going in a particular Ad</td>
<td>Noun</td>
<td>0</td>
</tr>
<tr>
<td>Communication traffic is the major problem in big cities.</td>
<td>Synset['traffic.n.03']</td>
<td>the amount of activity over a communication system during a given period of time</td>
<td>Noun</td>
<td>0</td>
</tr>
<tr>
<td>These two lanes merge on next traffic signal.</td>
<td>Synset['traffic.n.03']</td>
<td>the amount of activity over a communication system during a given period of time</td>
<td>Noun</td>
<td>0</td>
</tr>
<tr>
<td>Traffic jam caused by all the solitary car drivers.</td>
<td>Synset['traffic.n.03']</td>
<td>the aggregation of things (pedestrians or vehicles) coming and going in a particular Ad</td>
<td>Noun</td>
<td>0</td>
</tr>
<tr>
<td>Web links communication generate traffic for your site.</td>
<td>Synset['traffic.n.03']</td>
<td>the amount of activity over a communication system during a given period of time</td>
<td>Noun</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 6.4: Focus Word: Traffic

<table>
<thead>
<tr>
<th>SENTENCE</th>
<th>SYNSET</th>
<th>DEFINITION OF SYNSET</th>
<th>POS</th>
<th>SCORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>That was his turn to spin the wheel in the game.</td>
<td>Synset['turn.n.03']</td>
<td>the activity of doing something in an agreed succession</td>
<td>Noun</td>
<td>1</td>
</tr>
<tr>
<td>Please make a left turn from next signal.</td>
<td>Synset['turn.n.02']</td>
<td>the act of changing or reversing the direction of the course</td>
<td>Noun</td>
<td>1</td>
</tr>
<tr>
<td>Turn right at the junction to cross the bridge.</td>
<td>Synset['turn.n.01']</td>
<td>a circular segment of a curve</td>
<td>Noun</td>
<td>1</td>
</tr>
<tr>
<td>Turned out to be worse than any of us imagined.</td>
<td>Synset['turn.n.04']</td>
<td>a movement in a new direction</td>
<td>Verb</td>
<td>1</td>
</tr>
<tr>
<td>It is the turn of the next player to bat.</td>
<td>Synset['turn.n.03']</td>
<td>the activity of doing something in an agreed succession</td>
<td>Noun</td>
<td>1</td>
</tr>
<tr>
<td>Drive slow on the next turn.</td>
<td>Synset['turn.n.09']</td>
<td>a division during which one team is on the offensive</td>
<td>Noun</td>
<td>0</td>
</tr>
<tr>
<td>Next team will score higher when their turn comes.</td>
<td>Synset['turn.n.06']</td>
<td>the act of turning away or in the opposite direction</td>
<td>Noun</td>
<td>0</td>
</tr>
<tr>
<td>Please turn off the lights.</td>
<td>Synset['turn.n.09']</td>
<td>a division during which one team is on the offensive</td>
<td>Noun</td>
<td>0</td>
</tr>
<tr>
<td>I turn_down the job offer.</td>
<td>Synset['nonacceptance.n.01']</td>
<td>the act of refusing an offer</td>
<td>Verb</td>
<td>1</td>
</tr>
<tr>
<td>They turn_off the lights of the room.</td>
<td>Synset['toggle.n.02']</td>
<td>a side road where you can turn off</td>
<td>Verb</td>
<td>0</td>
</tr>
<tr>
<td>There is sharp turn ahead on the highway.</td>
<td>Synset['bend.n.01']</td>
<td>a circular segment of a curve</td>
<td>Noun</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 6.5: Focus Word: Turn

<table>
<thead>
<tr>
<th>SENTENCE</th>
<th>SYNSET</th>
<th>DEFINITION OF SYNSET</th>
<th>POS</th>
<th>SCORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>His mental state was not good when he decided to leave.</td>
<td>Synset['state.n.02']</td>
<td>the way something is with respect to its main attributes</td>
<td>Noun</td>
<td>1</td>
</tr>
<tr>
<td>Nation states grew into empires, with ever more power.</td>
<td>Synset['state.n.04']</td>
<td>a politically organized body of people under a single government</td>
<td>Noun</td>
<td>0</td>
</tr>
<tr>
<td>Water can be found in all the three states of matter.</td>
<td>Synset['state_of_matter.n.02']</td>
<td>(chemistry) the three traditional states of matter are solids (fixed shape</td>
<td>Noun</td>
<td>0</td>
</tr>
<tr>
<td>States of consciousness.</td>
<td>Synset['state.n.02']</td>
<td>the way something is with respect to its main attributes</td>
<td>Noun</td>
<td>1</td>
</tr>
<tr>
<td>State legislature should pass these laws also.</td>
<td>Synset['state.n.04']</td>
<td>a politically organized body of people under a single government</td>
<td>Noun</td>
<td>1</td>
</tr>
<tr>
<td>State of serious disrepair over the last few decades.</td>
<td>Synset['state.n.02']</td>
<td>the way something is with respect to its main attributes</td>
<td>Noun</td>
<td>1</td>
</tr>
<tr>
<td>State of the art labeling can place in a single process.</td>
<td>Synset['state.n.02']</td>
<td>the way something is with respect to its main attributes</td>
<td>Noun</td>
<td>1</td>
</tr>
<tr>
<td>This is not a citizenship bequeathed to people by a sovereign state.</td>
<td>Synset['state.n.02']</td>
<td>the way something is with respect to its main attributes</td>
<td>Noun</td>
<td>1</td>
</tr>
<tr>
<td>The patient’s mental state was not improved in the hospital.</td>
<td>Synset['state.n.02']</td>
<td>the way something is with respect to its main attributes</td>
<td>Noun</td>
<td>1</td>
</tr>
<tr>
<td>Nevada state’s population is almost equal to Wyoming’s.</td>
<td>Synset['state.n.03']</td>
<td>the territory occupied by one of the constituent administrative districts</td>
<td>Noun</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 6.6: Focus Word: State
Here we present the sentiment measure of people about current transportation system. Intensity of red color increases with the negative emotion and blue corresponds to positive emotion. Figure 6.9 shows the sentiment around the globe:
Figure 6.9: Sentiment Analysis Result Around the Globe

Figure 6.10: Sentiment Analysis Result Around the US
6.4 Conclusions

Our Tests resulted in 66.03% of 1’s as compared to score 0 which is nothing but the success rate of our algorithm. Thus we can predict the concept of any sentence with the information present in it taking the help of semantic dictionary like WordNet. Furthermore, no corpus is needed for disambiguate word senses in a sentence. This surely is a breakthrough which can result in quite successful automation algorithms for text, in future.

6.4.1 Advantages

Main advantage of our technique is that it doesn’t require any corpus and therefore no human effort is required to disambiguate words senses. It makes the system robust and quite easy to understand without any dependency on the amount of corpus. In this way, complexity of the system is also decreased.

The other advantage of this proposal is in the field of unsupervised document classification in general classes. Segmenting the document in sentences and looking into the concepts of each sentence we can find the context of document by looping the algorithm over those sentence contexts.

6.4.2 Disadvantages

This system highly relies on WordNet. It depends on the robustness of this semantic lexical dictionary. Advancement in WordNet network will directly improve the performance of our system. A lot of work is being done and going to be done to
improve WordNet in future.

POS like Verb, Adjective and Adverb do not have semantic relational trees that is there are not efficient ways to extract exact meaning of such POS. Although derivationally related forms provide a significant way to rectify this issue but it’s efficiency still lacks.

It depends on NLTK-POS tagging. Accuracy of POS tagging also effects the system vastly. More is the accuracy of POS tagging more will be the result percentage.

Other then these issues, there is an issue of calculation over all the combinations of synsets as we have to estimate the summation of WUP similarity for every possible combination of synsets which increase drastically with the number of words in the sentence. So, this algorithm is good for short sentences in which the evaluation is fast. It works for longer sentences as well but at the cost of more calculations.

6.4.3 Improvement Proposal

There are several possibilities for improvement in our system. One being the efficiency of Part of Speech tagging. Efficient POS tags can drastically improve the performance of this algorithm.

Moreover, We have not yet utilized the full power of WordNet. One important aspect which can further improve things is the handling of phrases. WordNet has beautifully arranged those phrases in the semantic network by joining their words with underscore. If Phrases can be identified within text, we can produce much improved results. One way to do that is to look for neighbouring words. For example if turn
has the word *down* ahead of it, algorithm should consider *turn_down* as a single word and search for its synsets. This improvement has been proven in the results shown.
BIBLIOGRAPHY


VITA

Graduate College
University of Nevada, Las Vegas

Himanshu Verma

Home Address:
3925 Cambridge ST Unit-2 Las Vegas, Nevada 89119

Degrees:
Bachelor of Technology, in Electronics and Communication Engineering, 2009
Indian Institute of Technology Guwahati, India

Thesis Title: Non-Learning Semantic Analysis for Context Discovery and Sentiment Estimation: Transportation Application

Thesis Examination Committee:
Committee Chair: Dr. Pushkin Kachroo
Committee Member: Dr. Mei Yang
Committee Member: Dr. Emma E. Regentova
Graduate Faculty Representative: Dr. Haroon Stephen