T.C.

BAHÇEŞEHİR UNIVERSITY

MULTI-VIEW SHORT-TEXT CLASSIFICATION USING KNOWLEDGE BASES

Master's Thesis

MERT ÇALIŞAN

İSTANBUL, 2016

T.C.

BAHÇEŞEHİR UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

COMPUTER ENGINEERING

MULTI-VIEW SHORT-TEXT CLASSIFICATION USING KNOWLEDGE BASES

Master's Thesis

MERT ÇALIŞAN

Thesis Supervisor: ASST. PROF. C. OKAN ŞAKAR

İSTANBUL, 2016

THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
COMPUTER ENGINEERING

Name of the thesis: Multi-View Short-Text Classification Using Knowledge Bases
Name/Last Name of the Student: Mert Çalışan
Date of the Defense of Thesis: 05.01.2016

The thesis has been approved by the Graduate School of Natural and Applied Sciences.

Assoc. Prof. Nafiz ARICA
Graduate School Director

I certify that this thesis meets all the requirements as a thesis for the degree of Master of Science.

Asst. Prof. Tarkan AYDIN
Program Coordinator

This is to certify that we have read this thesis and we find it fully adequate in scope, quality and content, as a thesis for the degree of Master of Science.

Examining Comittee Members	Signature
Thesis Supervisor Asst. Prof. Cemal Okan ŞAKAR	----------
Thesis Co-supervisor	----------
Member Asst. Prof. Görkem SERBES	----------
Member Asst. Prof. Tarkan AYDIN	----------

ABSTRACT

MULTI-VIEW SHORT-TEXT CLASSIFICATION
USING KNOWLEDGE BASES

Mert ÇALIŞAN

Computer Engineering
Thesis Supervisor: Asst. Prof. C. Okan ŞAKAR

January 2016, 50 pages

Automated text classification becomes more popular in recent years due to great increase in digitalization, content sharing and generation in the internet community. Machine learning algorithms are commonly used to classify various kinds of documents. Although the success of algorithms in document classification have been shown on various datasets from different domains, the traditional representation and classification approaches used to process normal-length documents fail in processing short-text messages such as customer reviews in e-shopping websites, personal updates in microblogging sites, or headlines in news portals. Therefore, there is an increasing need for more sophisticated algorithms to process short-texts. The traditional Bag-of-words representation when used for short-text documents results in very sparse data matrices that do not contain sufficient amount of information to obtain generalizable classification and clustering models. Besides, considering that millions of short-texts are generated every day, there is an increasing need for semi-supervised models to incorporate these unlabeled samples to the training phase. In this thesis, a semi-supervised learning model is proposed which is based on generating multiple views by enriching the short-texts using knowledge bases and then combining the predictions of these views to integrate the unlabeled samples to the training phase incrementally. An

experimental dataset consisting of Turkish short-text is used. The results show that the proposed method increases the accuracy compared to classical bag-of-words vector representation especially for small sample-sized training sets.

Keywords: Short-text Classification, External Knowledge Base, Semi-supervised Learning, Machine Learning

ÖZET

BİLGİ TABANI KULLANARAK ÇOK GÖRÜNTÜLÜ KISA METİN SINIFLANDIRMA

Mert ÇALIŞAN

Bilgisayar Mühendisliği
Tez Danışmanı: Yard. Doç. Dr. C. Okan ŞAKAR

Ocak 2016, 50 sayfa

Dijitalleşme, internet ortamında içerik paylaşımı ve üretiminin son yıllardaki büyük artışı, otomatik metin sınıflandırmanın daha popüler olasına sebebiyet verdi. Makine öğrenmesi algoritmaları, çeşitli tiplerdeki dokümanların sınıflandırılması için yaygın olarak kullanılmaktadır. Farklı alanlara ait çeşitli veri kümeleri üzerinde doküman sınıflandırma algoritmalarının başarısı gösterilmiş olsa da, normal uzunluktaki dokümanları işlemek için kullanılan geleneksel gösterim ve sınıflandırma yöntemleri e-ticaret sitelerinde bulunan müşteri yorumları, microblogging platformlarındaki kişisel paylaşımlar veya haber sitelerindeki manşetler gibi kısa metinlerin sınıflandırılmasında başarısız olmaktadır. Bu yüzden, kısa metinleri işlemek için daha sofistike bir algoritmaya duyulan ihtiyaç artmaktadır. Geleneksel kelime torbası gösterimi kısa metin dokümanlarına uygulandığında oldukça seyrek veri matrisleri ortaya çıkmakta ve bu gösterim genellenebilir sınıflandırma ve kümeleme modelleri elde etmek için yeterli miktarda bilgiyi bulundurmamaktadır. Aynı zamanda, her gün üretilen milyonlarca kısa metni dikkate aldığımızda, işaretlenmemiş bu verileri öğrenme fazında veri kümesine dahil etmek için yarı gözetimli öğrenme modellerine olan ihtiyaç artmaktadır. Bu tezde, kısa metinleri harici bilgi tabanı kullanarak zenginleştirip çoklu görüntü üreten ve bu görüntülerin tahminlerini, işaretlenmemiş örnekleri öğrenme fazına entegre etmekte

kullanan yarı gözetimli öğrenme modeli önerilmektedir. Türkçe kısa metinlerden oluşan deneysel bir veri kümesi kullanılmaktadır. Sonuçlar, önerilen metodun özellikle az sayıda örneğe sahip eğitim kümelerinde, klasik kelime torbası vektör gösterimine oranla başarıyı artırdığını göstermektedir.

Anahtar Kelimeler: Kısa Metin Sınıflandırma, Harici Bilgi Tabanı, Yarı Gözetimli Öğrenme, Makine Öğrenmesi

TABLE OF CONTENTS

TABLES ..... **ix**
FIGURES ..... **x**
ABBREVIATIONS ..... **xi**
1. INTRODUCTION **2**
1.1 PROBLEM STATEMENT **2**
1.2 PROPOSED SOLUTION **4**
2. LITERATURE REVIEW **7**
2.1 KNOWLEDGE BASE **7**
2.2 SHORT-TEXT CLASSIFICATION **9**
2.3 MULTI-VIEW TEXT CLASSIFICATION **10**
2.4 TEXT CLASSIFICATION IN TURKISH LANGUAGE BASED ON KNOWLEDGE BASES **11**
3. DATASET DESCRIPTION **13**
3.1 SEED SHORT-TEXT DATASET CREATION **14**
3.2 WIKIPEDIA REFERENCE KNOWLEDGE BASE GENERATION **16**
3.3 MULTI-VIEW SHORT-TEXT DATASET GENERATION **21**
4. METHOD **25**
4.1 SUPERVISED LEARNING **26**
4.2 FEATURE SELECTION AND EXTRACTION **30**
4.3 SEMI-SUPERVISED LEARNING **32**
5. RESULTS **35**
5.1 SUPERVISED LEARNING RESULTS **35**
5.2 FEATURE SELECTION AND EXTRACTION RESULTS **38**
5.3 SEMI-SUPERVISED LEARNING RESULTS **39**
6. DISCUSSION **43**
7. CONCLUSION **45**
REFERENCES **47**

TABLES

Table 3.1: Topic to class mappings…….…….....14
Table 3.2: Final seed short-text dataset characteristics……...……...……15
Table 3.3: Number of instances per class…….……...15
Table 3.4: Wikipedia category to class label mapping……..……..16
Table 3.5: CatScan configuration.....…….....…...17
Table 3.6: Article count per class label.....…….…..…....18
Table 3.7: Article entities and explanations.....……..……...19
Table 3.8: StringToWordVector filter, used parameter values…….....……24
Table 5.1: V1 supervised learning results……...…….……...35
Table 5.2: V2 supervised learning results ……..….……...36
Table 5.3: V3 supervised learning results ……....……..36
Table 5.4: Best supervised learning results compared..…….…....……....…….37
Table 5.5: Feature selection and extraction accuracies…….…..…….39
Table 5.6: Semi-supervised versus supervised learning..……...…….…….…..40

FIGURES

Figure 3.1: Class – Element – N Gram combinations…….20
Figure 3.2: Weighting pseudocode..……21
Figure 3.3: V1 and V2 distinction……...22
Figure 3.4: Evaluation process example……..23
Figure 4.1: Supervised learning overview……...27
Figure 4.2: SVM linear kernel parameter selection…….28
Figure 4.3: SVM RBF kernel parameter selection……..29
Figure 4.4: Basic ELM parameter selection……30
Figure 4.5: Discretization of V1……..31
Figure 4.6: mRMR features evaluation……...32
Figure 4.7: Semi-supervised learning……..34
Figure 5.1: Best supervised learning results graph…….…….37
Figure 5.2: mRMR feature selection graph…….38
Figure 5.3: V1 semi-supervised versus supervised……...……..41
Figure 5.4: V2 semi-supervised versus supervised……...……..41
Figure 5.5: V3 semi-supervised versus supervised……...……..42

ABBREVIATIONS

BoW : Bag of Words

ELM : Extreme Learning Machine

mRMR : minimum Redundancy Maximum Relevance

PCA : Principal Components Analysis

RBF : Radial Basis Function

SVM : Support Vector Machine

TFxIDF : Term Frequency - Inverse Document Frequency

V1 : View One

V2 : View Two

V3 : View Three

XML : Extensible Markup Language

1. INTRODUCTION

In this thesis, aim is to classify Turkish short-texts based on generating multiple views by enriching the dataset using knowledge bases. The thesis is organized as follows: In this section, the problem statement and suggested solutions are given in high level of detail. In section 2, literature review is given for usage of knowledge bases in machine learning tasks on text data, popular methods in short-text classification and text classification in Turkish language utilizing knowledge bases. In section 3, data collection, preprocessing and generation steps are described in detail under three subsections. These subsections are organized as collection and preprocessing of seed short-text dataset, extraction and preprocessing of Wikipedia articles and generation of reference knowledge base and finally generation of multiple views via using seed short-text dataset and reference knowledge base. In section 4, the methods used in classification phase are given as supervised learning, feature selection and extraction and semi-supervised learning combined with ensemble learning. Section 5 gives the experimental results for application of methods described in section 4. In section 6, discussion of the experiment results are given. Finally, study concluded in conclusion section.

1.1 PROBLEM STATEMENT

In recent years, there is a great increase in digitalization, content sharing and generation on web. This increase in the number of digital text content fueled the need for automated document and text classification. In the modern era of natural language processing, this problem is tried to be solved with machine learning. The supervised machine learning algorithms are trained on the training dataset consisting of manually labeled samples, and the model optimized on validation set is applied on the test set to report the final accuracy. The success of machine learning classifiers depends highly on the quality and quantity of the samples used. In case of text classification problem, the dataset consists of manually labeled documents. Considering the huge amount of text

data generated every day especially on web, the necessity of semi-supervised learning algorithms that automatically label the documents to incorporate into the training phase is increasing.

In this approach, typically a small amount of manually labeled data is used to label a large amount of unlabeled data to use in the training phase. This approach fits very well to short-text classification problem since it is very easy to obtain unlabeled data by crawling messages and comments from the Web but labeling these samples requires heavy manual effort resulting slow human annotation.

The feature extraction step is also one of the most important tasks that should be addressed in short-text classification problems. The traditional approach used to represent documents is called Bag-of-Words (BoW), in which each document is represented with a vector of words where word occurrence based weights are used as feature values (Gabrilovich and Markovitch 2006, Sebastiani 2002). BoW representation of text is the most commonly used method in document classification problems (Gupta and Lehal 2009). However, the sparsity of data matrix obtained, when short-texts including only a few words are represented with BoW, becomes a key problem for classification task (Man 2014, Wang et al. 2012). This problem is getting more important with increasing interest of users in services like the microblogging and social media. There are some research efforts that enrich the content of short-texts as a preprocessing step and then feed the enriched dataset to machine learning classifiers to increase the generalization and accuracy of the final model.

Since short-text typically consists of a few words, BoW representation of short-text results in a very sparse vector. One of the main reasons of low accuracy in statistical classifiers is this sparseness (Wang et al. 2012). Also not only for short-text classification but in general, BoW is criticized because of its lack of carrying semantic knowledge. Vector of terms does not contain the order of term occurrence and evaluates terms as they are independent which makes loss of a basic semantic knowledge (Huang et al. 2008).

Researchers' solutions to BoW limitations on the short-text classification include both using different representations other than BoW and extending BoW. In both ways, a widely used technique is to utilize an external knowledge taken from an external data source. Either sample is represented in the domain of this external knowledge or BoW extended with new features extracted from external knowledge. External knowledge is not only used for statistical information but also used as a semantic information source with its domain specific properties like titles and links of a web page.

It should be noted that using external sources to construct a knowledge base with high quality and wide coverage requires serious processing effort for cleaning the data. In addition to this, some of the related methods proposed in the literature are not very suitable for real time tasks and scenarios due to their high computational complexity (Wang et al. 2012).

To sum up, there is a need for an accurate and scalable short-text classification method that can be used in real time applications. Considering that labeling the huge amount of text generated on web requires heavy human annotation effort, the proposed solution should address learning generalizable model with small sample-sized training data.

1.2 PROPOSED SOLUTION

Overcoming the problems of short-text classification described in the problem statement section requires a solution with various methods. In this thesis research, aim is to propose a model that addresses the problems stated about short-text classification. The dataset used in this thesis is composed of 6 classes. Short-text samples in this dataset contain 146 characters on average. The dataset is in Turkish language about which there are far less number of studies when compared to English.

In this thesis, to deal with the sparseness problem of BoW and lack of semantic information, an external knowledge is used. Gabrilovich and Markovitch (2006) stated that encyclopedic knowledge is both useful for classifying short documents and when

there are not enough number of samples. Wikipedia (Vikipedi for Turkish)[1] is the greatest digital and free encyclopedia in Turkish. It is well-structured and available on the internet. Due to these reasons, Wikipedia is used as an external knowledge in this thesis.

For each class in dataset, relevant category in Wikipedia is found and articles in that category are selected according to their depth. Wikipedia has a tree like category and article structure, and articles only above a certain depth are taken for relevance and quality concerns. Articles are processed and used to construct a reference knowledge base where data is stored in a database structure that enables quick referencing for classification of short-text samples in real world. This knowledge base contains also semantic information which is generated using domain specific properties of Wikipedia articles like anchor texts, titles, image descriptions, and many others. Zhang et al. (2013) proposed that domain specific properties of Wikipedia like titles have important semantic value.

In this thesis, each short-text sample in the dataset is represented with three views. First view is the Wikipedia domain specific representation of short-text sample. As an example, one of the features constructed for this view represents the weight of terms in the short-text sample when terms are considered with titles of articles which belong to Biology category in Wikipedia. Second view is the Wikipedia article text representation of short-text sample which is constructed with the same approach used in first view except only the article texts are considered instead of domain specific properties of Wikipedia. While first view requires an external source with specific properties, an external source only with plain text is enough for second view. Thus, aim is to compare the effectiveness of domain specific properties and plain texts obtained from external sources. Gabrilovich and Markovitch (2006) stated that "…the correct sense of each word is determined with the help of its neighbors". For this reason, along with the domain specific properties in the first view, n-gram based features are also generated and used to support semantic information for both views.

[1] https://tr.wikipedia.org

As the third view, traditional BoW representation of short-text sample is used with Term Frequency - Inverse Term Frequency (TF x IDF) weighting scheme. Each view is first fed to the classifier individually. Support Vector Machine (SVM) (Cortes and Vapnik 1995) and Extreme Learning Machine (ELM) (Huang et al. 2006) are used as classifiers due to SVM's successful applications in text classification studies and ELM being a new promising classifier. Feature selection using minimum Redundancy Maximum Relevance (mRMR) (Sakar et al. 2012, Peng et al. 2005) filter feature selection method and feature extraction using Principal Component Analysis (PCA) are also applied to improve the accuracy of the classifiers.

The individual predictions of views are combined using voting and the samples that the views most agreed upon are incorporated to the training phase. This multi-view semi-supervised learning approach is compared to supervised learning on a Turkish short-text dataset.

In conclusion, Wikipedia is used as an external data source to create a knowledge base. Two new views of short-text sample are generated using the knowledge base with acceptable extra computational load to be used in real time applications. Domain specific properties of Wikipedia are used for semantic information in one view while n-gram representation features are used in both views for additional semantic information. In addition to the generated two views, traditional BoW representation view of short-text is also constructed, and the individual predictions of the views are combined to obtain new labeled examples to increase the number of samples used in the training phase. Thus, the short-text classification problem with limited number of labeled samples is addressed.

2. LITERATURE REVIEW

Related literature studies are given under 4 subsections. In knowledge base section, different types of research done on mining of external data sources for extracting knowledge to support classification of samples are given. Short-text classification section contains the most popular techniques for resolving the short-text classification problem. Multi-view text classification section contains multi-view approaches in text classification. Finally, last section gives information about research done on Turkish text classification using Wikipedia.

2.1 KNOWLEDGE BASE

The general notion is to extract data from the external data source to build a knowledge base, and then to enrich the samples in dataset with new features or new samples using this knowledge base.

As a common method, a sample is searched within the knowledge base using a text similarity measure to find a relevant data for enriching the sample (Zhang et al. 2008). Gabrilovich and Markovitch (2006) built an auxiliary text classifier which is able to find the most relevant Wikipedia articles for samples. The determinate articles are used to extend BoW representation of samples with new features.

The recent studies show that enriching with using semantic properties of text or data source is a successful method. For this method, knowledge bases with ontologies are preferred for their high amount of semantic data. Rafi et al. (2012) used Wikitology, Wikipedia with ontology, for sample enrichment where samples are searched semantically in Wikitology. Entities and entity types in the samples are identified and used in semantic search to find right and valuable information for enrichment. As a result, it was shown that enrichment with semantic methods outperforms enrichment without semantic methods.

Although a data source with ontology is a very valuable asset, extracting semantic information does not require ontology. Properties of data sources also provide semantic information. An example to those properties is anchor texts in Wikipedia articles (Huang et al. 2008, Gabrilovich and Markovitch 2006). Other than properties, even plain text itself contains semantic information like sequence of terms which can be represented in n-gram representations for their semantic value (Huang et al. 2008).

Knowledge bases are also used for automated labeling of samples where there are not enough samples for classification experiment. As an example, Zhang et al. (2013) used Wikipedia for label propagation from labeled samples to unlabeled samples. Semantic information and properties of Wikipedia are used to construct combined similarity measures. These measures are used for clustering, and then according to clustering result, the labeled samples are used to label unlabeled samples which are closer to centroid of cluster. Remaining unlabeled samples and new training samples are used with transductive SVM for semi-supervised classification. Zhang et al. (2013) also stated that the number of research done for semi-supervised learning with utilizing Wikipedia knowledge is few.

Another useful method of utilizing knowledge bases is bootstrapping approaches in which instead of using a labeled training set, only important keywords describing classes are given. Zhang et al. (2008) designed a framework called Knowledge Supervised Learning where keywords are used to extract information from data sources like Wikipedia, and then extracted information used for classification. Their results are comparable to supervised learning with SVM classifier using labeled dataset.

Usages of knowledge bases are not limited to classification tasks; they are used for clustering tasks as well (Huang et al. 2008). Also many methods benefit from clustering for finding relevant information from knowledge base to be used in classification tasks (Zhang et al. 2008).

2.2 SHORT-TEXT CLASSIFICATION

According to the Wang et al. (2012), there are three popular short-text classification methods at present. One is classification based on similarity measures of terms where similarity is calculated using search engines and measure is used in classification. This method increases the accuracy to some extent but requires frequently querying of search engines which is not efficient for real time tasks due to performance concerns. Another way is using a knowledge base to extend samples' features or map terms of samples to knowledge base articles or topics. This method also increases the accuracy but due to heavy volume of knowledge base data, requires a lot of data processing for elimination of noise and unrelated terms. An alternative method is mining text data based on topics to find co-occurrence probability of terms in short-text samples in a topic which also increases the accuracy to some extent.

Wang et al. (2012) collects balanced and rich external text data for each class. Using latent drichlet algorithm and information gain, important terms for each class are selected and terms with high multi-class probabilities are filtered. A thesaurus is constructed with selected terms. After preprocessing of short-text samples, feature terms in the samples are assigned to weights with using thesaurus. The resulting sample dataset is used in classification.

Man (2014) aimed to enhance word vector representation in short-text classification problem. He uses news data as external information source and makes association rule mining on it to identify double term sets. These term sets are identified based on co-occurrences and same class orientation of terms. Identified double term sets are used as a knowledge base to make feature extension on original word vector. Each term in the original sample is searched in double term sets to find its pair and add to original word vector form.

One of the most important areas that short-text classification used is social media and micro blogging services where user generated short-text contents are huge. BoW model

limitations are tried to be resolved with the support of features generated from domain specific properties like profile of the author and its previously generated content (Sriram et al. 2010). These properties hold semantic value just like properties of knowledge bases described in previous section. Advanced NLP processing is required since the content generated is mostly contains slang words, emoticons and shortenings. While these slang and shortened terms may cause ambiguities, they also have an important value for discrimination.

2.3 MULTI-VIEW TEXT CLASSIFICATION

Mostly, multi-view text classification is applied with combining semi-supervised learning approach. Multiple views of text are constructed for semantic representation, syntactic representation and text representation. Some methods use samples alone to form different view representations while others utilize an external knowledge.

Matsubara et al. (2005) proposed an approach to generate different views of text. Two view representation of text generated which first view is 1-gram term representation and second view is 2-gram term representation and values of features are term frequencies. A separate classifier, based on Naïve Bayes, for each view is trained in a supervised learning manner. Trained classifiers are used to label unlabeled samples with assigning confidence level where samples with high confidence are added to training set until a termination criteria is reached.

Active semi-supervised learning method based on Naïve Bayes classifier for large number of unlabeled data is proposed by Gu et al. (2009). In the proposed method, three views of text; a BoW based lexical view, word to concept mapping based semantic view which an external knowledge base is utilized for concepts and syntactic view with syntactic features like parts-of-speech tags are generated. A separate classifier for each view is trained. Unlabeled data is selected to be used in training set with an uncertainty measure, confidence and majority voting of view classifiers. Classifiers are

incrementally updated with selected samples until no unlabeled sample left with qualification to be selected.

Li et al. (2012) aimed to improve the performance of transductive SVM and make use of unlabeled samples for less labeling effort. A multi-view semi supervised learning approach is proposed where two semi-supervised learning techniques are utilized together; exploring manifold structure and maximizing margin. A separate classifier for each view is trained and in training, penalty is applied to the decision function of each classifier based on consensus in results. Proposed method is evaluated with binary classification on different types of multi-view datasets. Specifically, method performed on two view product reviews dataset where one view is word vector model representation of text and other view is built up of features like length of text and numerical digit proportion that represent properties of sentence. Another text dataset used has text content of a web page in one view while anchor texts pointing to that web page from other pages are placed in other view. Proposed method is proved to be better than single-view learning methods especially when the training set size is small.

2.4 TEXT CLASSIFICATION IN TURKISH LANGUAGE BASED ON KNOWLEDGE BASES

To the best of knowledge, there is only one study which utilize Turkish Wikipedia (Vikipedi) to enhance text classification in Turkish language (Poyraz et al. 2012). In this study, titles of Wikipedia articles are extracted and searched in sample documents. If a title is found in a sample document, then it is added as a single term, whatever the title's length is (i.e. contains 3 words), to the word vector form of sample document. It is stated that, by this way the semantic information in title is kept and used in classification. As a result, classification accuracy is improved compared to traditional BoW of sample documents. Also it is concluded that Turkish Wikipedia contains most of the terms in used sample dataset which is composed of Turkish newspaper articles.

Other than text classification, Turkish Wikipedia is utilized for text mining tasks like Turkish text summarization. Guran et al. (2013) used links in Turkish Wikipedia for

extracting semantic features and combine them with document’s structural features for Turkish text summarization.

3. DATASET DESCRIPTION

All of the data used in this research is in Turkish language. Multiple data sources and many extraction and preprocessing tasks are used to create final multi-view dataset which is used as input to the classifiers.

A text data is taken from TS Abstract Corpus[2] in which each sample belongs to one of the 6 classes; biology, sport, economics, technology, politics and religion. A dataset consisting of short-texts is then generated from the corpus data. Short-text instances were manually reviewed and checked if they belong to the tagged classes. Due to shortage of short-text data in some classes, dataset is enriched with short-text data that is extracted from 42 Bin Haber dataset (Yıldırım and Atık 2013). After completing these steps, seed short-text dataset is created.

Articles, which belong to the Wikipedia categories that are identical with 6 different classes of short-text dataset, are exported from Wikipedia. Exported articles are processed with natural language processing techniques, wiki markup processing and decomposed into a reference knowledge base with statistics, according to Wikipedia article features.

Final dataset is multi-view short-text dataset which is the representation of seed short-text in multiple views. There are three different views in the final dataset. First and second views are the representations of seed short-text dataset according to the reference knowledge base created from Wikipedia articles. These two views do not contain any feature terms like BoW since BoW representation has a limited performance especially on short documents. On the contrary of first two views, third view is the classical BoW word vector representation of seed short-text dataset. Natural language processing techniques are also applied to seed short-text dataset at the creation of multiple views.

[2] http://tanersezer.com/?p=203, accessed at 08/23/2015

In this chapter, each step of data processing is described in detail with statistical information.

3.1 SEED SHORT-TEXT DATASET CREATION

TS Abstract Corpus is a corpus in Turkish language which provides many features including data with topic tags. From tagged topics, 6 categories are chosen for classification, and data belonging to these categories is extracted. Class labels used in this thesis are not exactly the same with the topic name. In Table 3.1, TS Abstract Corpus topic tags and their respective class labels used in this thesis are given.

Table 3.1: Topic to class mappings

TS Abstract Corpus Topic Tag	**Class Label**
Biology	Biology
Sports	Sport
Economics	Economics
Information Management	Technology
Political Sciences	Politics
Religion	Religion

Instances (lines) in the extracted data are in the short-text form by default and have an id feature which identifies the source document of instance. The tags are converted to class labels. Most of the instances in the extracted data are not complete sentences. Each instance represents a small portion of its source document. It is identified that classifying some of the instances are not possible even for a person. Also it is found that some instances are belonging to the same source document and they are successors or predecessors of each other which may cause inaccuracy in the results. Due to these reasons, all of the instances are evaluated manually, ambiguous instances are removed and only one instance is kept from a single source document. Economics and Politics classes had more ambiguous instances compared to other classes. In order to have

enough data to make experiment, additional short-text samples have been manually extracted out of another data source, 42 Bin Haber dataset (Yıldırım and Atık 2013) for these classes. Additional short-text samples are similar to TS Abstract Corpus instances in length and sentence completeness.

After completion of the steps mentioned above, final seed short-text dataset is obtained. Note that, most of the instances are not full sentences also in the final dataset. The final seed short-text dataset has characteristics as mentioned in Table 3.2. Also number of instances per each class in dataset is given in Table 3.3.

Table 3.2: Final seed short-text dataset characteristics

Property	**Value**
Minimum number of characters per instance	66
Maximum number of characters per instance	220
Average number of characters per instance	146

Table 3.3: Number of instances per class

Class	**Number of Instances**
Biology	147
Sport	181
Economics	207
Technology	182
Politics	155
Religion	120
Total number of instances	992

3.2 WIKIPEDIA REFERENCE KNOWLEDGE BASE GENERATION

Other than providing encyclopedic information to end users, Wikipedia is a popular source of creating a knowledge base to use in the fields of natural language processing and machine learning. In this thesis, a representation of the short-texts is generated in Wikipedia domain to gain both semantic information and deal with sparseness of the traditional BoW representation. Since the short-texts are in Turkish language, Turkish Wikipedia is used. Note that, Wikipedia is a fast growing encyclopedia which means that the statistical information provided in this study may change in time. Due to this reason, dates are provided where needed.

As a first step, categories which are mapped to class labels are identified manually. Wikipedia category to class label mapping can be seen in the Table 3.4.

Table 3.4: Wikipedia category to class label mapping

Wikipedia Category	Class Label
Biyoloji	Biology
Ekonomi	Economics
Siyaset	Politics
Din	Religion
Spor	Sport
Teknoloji	Technology

Using a tool hosted on Wikipedia site, CatScan[3] version 2, a list of **articles** in these categories is identified. While using the CatScan, configuration in Table 3.5 is used. Default values are used for parameters which are not mentioned in the Table 3.4.

[3] https://tools.wmflabs.org/catscan2/catscan2.php, accessed at 08/23/2015

Table 3.5: CatScan configuration

Parameter	Value
Language	tr
Depth	1
Category	Category Name
Negative Category	Other Categories Name
Combination	Subset
Namespace	Article (only)

One of the most important parameters is the Negative Category. While inputting category name for a class given in Table 3.4, other classes' respective category names are placed in the negative category. The reason behind this is having a clear distinction between classes. An article in Wikipedia can be in more than one category at the same time and using an article with multiple categories that map to class labels will cause problem in classification. To deal with this problem and have a clear distinction between classes, negative category parameter is used.

Wikipedia has a hierarchical treelike category structure in which an article can be placed under multiple categories at the same time. Another important parameter is the Depth parameter for finalizing the search at the given depth. An article is in the depth 0 for a given category, if it is direct child of it. If an article is a direct child of the subcategory for a given category, then this article is in depth 1. It is observed that, with increasing depth, the articles become less relative to the given category. This might be because of either Turkish Wikipedia article quality or a result of Wikipedia category hierarchy which an article can be under multiple categories at the same time. Due to the relevance concern, depth is given as one. CatScan outputs the list of article names for the given parameters. In Table 3.6 you can see the number of articles per class label. CatScan is used at date August 23, 2015. Both statistics and results are relevant to that date.

Table 3.6: Article count per class label

Class Label	Article Count
Biology	216
Sport	374
Economics	682
Technology	407
Politics	599
Religion	212

After having list of articles for each class from CatScan, Wikipedia Export[4] tool is used to export these articles in Extensible Markup Language (XML) format. Exported data contains both metadata information and article information. The baseline Wikipedia article data used in this research is exported at date August 23, 2015. Both statistics and results are relevant to that date.

Exported articles in XML format are parsed by a developed sax parser in Java. The article text in the exported XML contains Wiki Markup Language tags. These markup tags define both visual rules and Wikipedia related (domain specific) properties of articles which provide semantic information. These rules and properties are named as **entities** in this research. Wiki Markup Language is a complex markup language due to diversity of tag usage it enables. These properties include some important entities like internal links to other articles, subtitles, image descriptions and many others. In this research, not only extracting the articles' plain text with clearing the markups done but also these entities defined with markups are also captured. To extract this knowledge, a regular expression based wiki markup parser is developed in Java. Although there are many open sourced wiki markup parsers developed in Java, none of them has the full capability to distinguish the aimed entities and capture desired information. In Table 3.7, extracted entities and their explanations are given.

[4] https://tr.wikipedia.org/wiki/%C3%96zel:D%C4%B1%C5%9FaAktar, accessed at 08/23/2015

Table 3.7: Article entities and explanations

Entity Name	Explanation
Internal Anchor Texts	Displayed text of links to another article
External Anchor Texts	Displayed text of links to other websites
Category Texts	Categories' text the article belongs to
Subtitle Texts	Headings' text excluding main heading
Brackets Texts	Used for other or relative meanings of a word also used for informing additional sources about article
Highlighted Texts	Bold, italic, underlined and big texts
Image Texts	Descriptions of Images

Article Title is also extracted but not from markup processing, instead it is taken in the XML parsing because it is defined separately from article body. Other than the entities, articles' plain texts are extracted. Articles plain text contains articles text except main heading, tables, references and strikethrough texts.

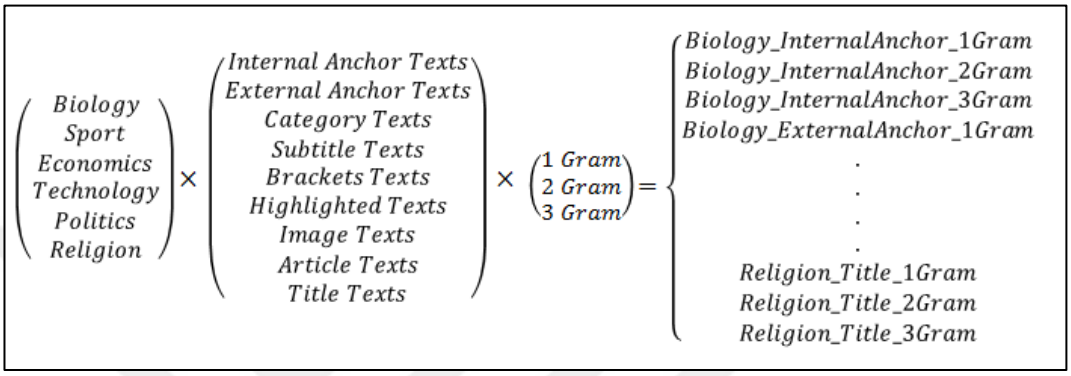
Extracted information is processed with natural language processing tasks and then stored in a structure which provides a quick referencing at creation of views of a sample, for using effectively in real world scenario. First Zemberek[5] is used for lemmatization process of words. Also dates in many formats are represented with a "dateobject" text and numbers are represented with a "numberobject" text to have a unified knowledge. Then word tokenization is applied to lemmatized texts with a custom tokenizer developed in Java. Tokenization outputs three different n-gram representations of texts with values 1, 2 and 3 for N.

Storage structure is composed of files, which can also be stored as database tables, containing lemmatized word tokens and counts and weights of them. There is one file for each Class – Element – N-Gram combination, where class is class label, element is an entity, title or article plain text and N's value is 1, 2 or 3. In Figure 3.1, possible combinations can be seen.

[5] https://github.com/ahmetaa/zemberek-nlp, accessed at 03/01/2015

For instance, Biology – Title – 1-Gram file contains the lemmatized 1-gram tokens, which means single words, counts, and weights. A token's count means; number of occurrences of this token in titles of articles, in biology category. Same principal is applied to all combinations.

Figure 3.1: Class – Element – N-Gram combinations



Count of tokens is not the main feature in the reference knowledge base. Instead, it is the output of an intermediate step, which supports to calculate the primary feature of a token; **weight**. Weight is the value calculated with a variation of weighting scheme Term Frequency - Inverse Document Frequency (TF x IDF) for identifying how much a token is important to that Class – Element – N-Gram combination. This is important because the number of articles per class and each article's length is not equal, which makes term frequency alone not a valid weighting scheme. The difference of used weighting scheme compared to standart TF x IDF scheme is instead of document, Element – N-Gram combination is considered. As a result, inverse document frequency is transformed to inverse term frequency calculated based on Element – N-Gram combination. This is because, in classical methods, interpretion of a document is a single typed entity containing text. In this research, Element – N-Gram combination results 27 different types of documents containing text to provide semantic information. For this reason, a token has a seperate inverse term frequency value for each Element – N-Gram combination, which is calculated via evaluating token only within that specific Element – N-Gram combination. Similarly, for each Class – Element – N-Gram

combination, a seperate term frequency for a token is calculated internally in that combination. In Figure 3.2, weighting scheme details can be seen as.

Figure 3.2: Weighting pseudocode

```
procedure weight(token,Class, Entity, Ngram)
term frequency ← term frequency(token, Class, Entity, Ngram)
inverse term frequency ← inverse term frequency(token, Entity, Ngram)
weight = term frequency * inverse term frequency
return weight

procedure term frequency(token, Class, Entity, Ngram)
count ← token's count value recorded in Class+Entity+Ngram file
total count ← total of all tokens' count values recorded in Class+Entity+Ngram file
tf ← log_e(count / total count)
return tf

procedure inverse term frequency(token, Entity, Ngram)
count ← total of token's count values recorded in Entity+Ngram files of all classes
total count ← total of all tokens' count values recorded Entity+Ngram files of all
classes
itf ← log_e(total count / count)
return itf
```

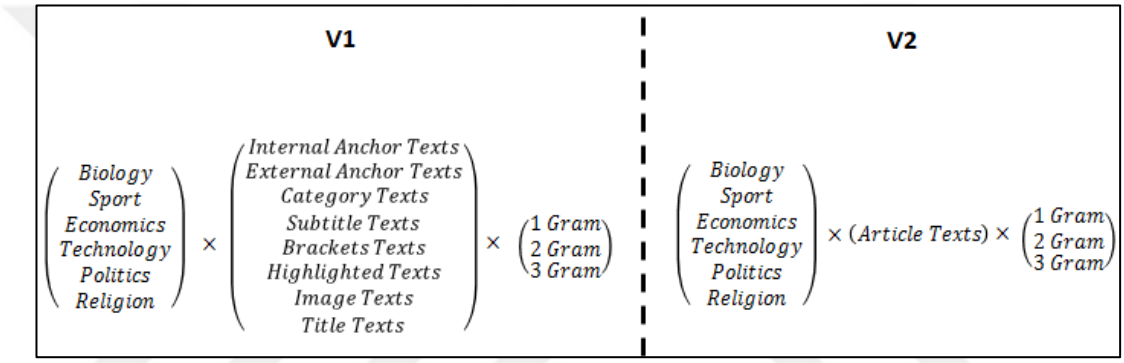
3.3 MULTI-VIEW SHORT-TEXT DATASET GENERATION

All short-text instances in the seed short-text dataset are processed to be represented in 3 different views. View One (V1) and View Two (V2) are generated according to the reference knowledge base which is built based on the Wikipedia articles. They do not contain any feature terms like BoW because aim is to overcome its limitations which are sparseness and lack of semantic knowledge. However, View Three (V3) is a word vector representation, BoW, generated from seed short-text dataset alone.

In section 3.2, it is stated that the structure of the reference knowledge base is constructed in a way to enable quick referencing while creating views of a sample. It is an important point to make use of this method in real world scenario. Structure enables quick referencing mainly because each Class – Element – N-Gram combination, which are files in reference knowledge base, is a feature in V1 or V2. V1 has 144 features, excluding class label feature, which are Class – Element – N-Gram combinations in

reference knowledge base where element is an **entity** or **title**. V2 has 18 features, excluding class label feature, which are Class – Element –N-Gram combinations in reference databese where element is **article plain text**. The distinction between V1 and V2 is, V1 represents Wikipedia domain specific view with features of entities and title which adds semantic information to representation, while V2 represents a generic view with features of article plain text which can be extracted also from **another source** with higher quality and plain text. Also both views include n-gram features which adds semantic information. In Figure 3.3, V1 and V2 distinction can be seen.

Figure 3.3: V1 and V2 distinction



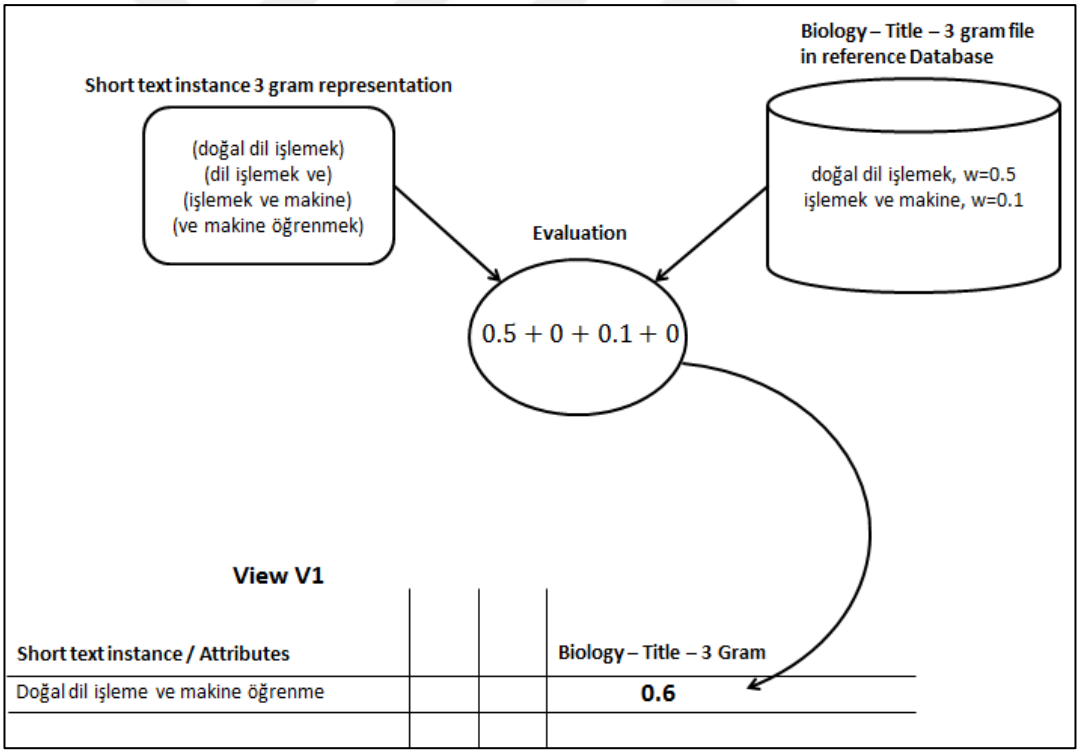
When a seed short-text instance is evaluated with a file in reference knowledge base, this outputs the value of the respective feature in V1 or V2. But first, natural language processing tasks are applied to seed short-text instances as it was applied for extracted Wikipedia texts while creating the reference knowledge base. Using Zemberek, lemmatization is done to words, dates, and numbers processed to be represented as "dateobject" and "numberobject" for having a unified knowledge. Then word tokenization is done with custom tokenizer developed in Java and 3 different n-gram representations of short-text are outputted for N values of 1,2 and 3.

A short-text instance is evaluated with each Class – Element – N-Gram combination file seperately in reference knowledge base. When evaluating, suitable n-gram representation of short-text is used according to the reference file. For instance, when evaluating the short-text with any Class – Element – 3-Gram combination, 3-gram

representation of short-text instance is used. The output of this evaluation is the feature value of short-text instance's respective Class – Element – N-Gram combination. In case, short-text instance is evaluated with Sport – Image Description – 2-Gram file in reference knowledge base, the output is the value of Sport – Image Description – 2-Gram feature in view V1 of short-text instance.

Evaluation of short-text instance with a Class – Element – N-Gram combination file is querying the tokens of short-text instance in Class – Element – N-Gram file so that weight values are taken for tokens. Tokens not existing in the file has 0 weight. Sum of short-text instance's tokens' weight values in Class – Element – N-Gram file is the output of evaluation. In Figure 3.4, an example of evaluation process is given.

Figure 3.4: Evaluation process example



After completion of evaluation process for each short-text instance with each Class – Element – N-Gram combination in reference knowledge base, V1 and V2 views are created.

Data mining and machine learning software WEKA (Hall et al. 2009) is used for most of the experiments in this thesis study. One of the functionality being used is the unsupervised filter RemoveUseless, to remove features of V1 and V2 which have 0 or over 99 percent variance. As a result of application of this filter, V2 is not affected and lost none of its features. However, **29 features** of V1 are removed using this filter, and the remaining 115 features are used in this representation. Removed features are the ones with no value as a result 0 variance and 5 of them are 2-gram combination while 24 of them are 3-gram combination. There are 48 features for each n-gram combination in V1 which means removed 3-gram features due to 0 value is the half of the 3-gram features in V1.

Another use of WEKA in data generation is the unsupervised filter StringToWordVector, to convert seed short-text dataset into word vector form which is the V3 representation. Output of the filter had 3565 features (tokens) with values generated according to TF x IDF weighting scheme. In Table 3.8, used values for parameters of StringToWordVector are given, where parameters used with default values are not mentioned.

Table 3.8: StringToWordVector filter, used parameter values

Parameter	Value
IDFTransform	True
TFTransform	True
wordsToKeep	10000 (to make sure it keeps all the tokens as features)

4. METHOD

There are two main classification methods used in this thesis research; supervised and semi-supervised learning. Supervised learning is done for each view separately and their accuracies are proposed. For V1, feature selection and feature extraction methods are applied and evaluated with supervised learning. Multi-view semi-supervised learning is applied with ensemble learning by combining the predictions of the views.

As classifier, SVM (Cortes and Vapnik 1995) and ELM (Huang et al. 2006) are used. SVM seperates samples from different classes in space with hyperplanes. For better generalization, SVM uses support vectors to maximize the margin between hyperplane and closest samples to the hyperplane. SVM is most widely used classifier for text classification due to its ability to handle high dimentionality and sparseness which appears also in vector space representation of text. ELM is a new kind of feedforward neural network algorithm which uses single hidden node layer. Unlike traditional neural network algorithms, ELM is both easier to use and extremely efficient in terms of computation. In ELM, weights between hidden and input nodes are never updated after randomly assigned and weights between hidden and output nodes are learned in a single iteration. Since there is no multiple iterations for learning like traditional neural network algorithms, ELM is very fast at learning. ELM only takes number of hidden neurons (nodes) as input while traditional neural network algorithms take many input parameters to tune the network which make ELM better at avoiding local optimal solution and reaching high generalization performance (Ding et al. 2015).

SVM is selected to be used for semi-supervised learning due to its superiority to ELM in supervised learning experiments. The LibSVM (Chang and Lin 2011) implementation via wrapper of WEKA software is used for SVM classification while open source java implementation[6] of ELM is used. The classification procedures are repeated 10 times for statistical significance with randomly generated training and test

[6] http://www3.ntu.edu.sg/home/egbhuang/source_codes/ELM-Java.zip, accessed at 09/01/2015

sets, and the average accuracies obtained on test sets are reported. For feature selection, mRMR (Sakar et al. 2012, Peng et al. 2005) is used and for feature extraction PCA in WEKA is used. mRMR feature selection tries to identify and select the features that are most effective for defining the class considering two constraints; minimum redundancy in terms of features similarity to each other and maximum relevance in terms of features similarity to class feature. PCA feature extraction tries to map the data to a lower number of dimensions according to the variance of features. Rather than selecting features, new features are extracted from existing ones in PCA. In the following subsections, details of how the methods used for experiments are described.

4.1 SUPERVISED LEARNING

Supervised learning experiments are done with training set sizes 80, 60, 40 and 20 samples **from each class**. All training samples are selected randomly and the rest of the samples in the dataset are used for testing. The training samples are chosen such that the class distribution in the training set is equal. Classifiers' parameters are chosen with trial of a set of values' on test set and the parameters that yield the best accuricies are determined. All experiments are repeated 10 times and average test set accuracies are reported. In Figure 4.1, the overview of the supervised learning process is given as pseudocode.

Figure 4.1: Supervised learning overview

```
procedure supervised learning(x)
accuracy ← 0
 for i ← 1 to 10
     training ← select x samples randomly for each class
     test ← rest of the samples
     accuracy ← accuracy + classifier(training,test)
 next i
accuracy ← accuracy / 10
end

procedure classifier(training,test)
accuracy ← 0
best accuracy ← 0
 for each pval in parameter values
    accuracy ← classification(pval,training,test)
    if best accuracy < accuracy then
       best accuracy = accuracy
    end
 end
return best accuracy
end
```

One classifier used is LIBSVM implementaion of SVM which is used via wrapper of WEKA machine learning software. For each view, C-SVC SVM type, linear and radial basis function (RBF) kernels are tried. For linear kernel, cost parameter value is choosen as shown in Figure 4.2. It is started from 0.5 and multiplied by 2 in the end of each iteration till reaching 512 and the parameter corresponding to the highest accuracy is chosen.

Figure 4.2: SVM linear kernel parameter selection

```
procedure svm linear(training,test)
accuracy ← 0
best accuracy ← 0
cost ← 0.5
while cost < 512
    accuracy ← classification(cost,training,test)
    if best accuracy < accuracy then
      best accuracy = accuracy
    end
    cost ← cost x 2
 end
return best accuracy
end
```

For RBF kernel, same parameter optimization approach is used to determine the cost parameter. After determining the best c value, then gamma values are tried with best c value. Gamma starts with value 0.0005 and multiplied by 2 in the end of each iteration till reaching maximum value below 1. In Figure 4.3, RBF parameter selection pseudocode is given.

Figure 4.3: SVM RBF kernel parameter selection

```
procedure svm rbf(training,test)
accuracy ← 0
best accuracy ← 0
cost ← 0.5
best cost
gamma ← 0.0005
while cost < 512
    accuracy ← classification(cost, gamma, training,test)
    if best accuracy < accuracy then
       best accuracy ← accuracy
       best cost ← cost
    end
    cost ← cost × 2
end
while gamma < 1.0
    accuracy ← classification(best cost, gama, training,test)
    if best accuracy < accuracy then
       best accuracy ← accuracy
    end
    gamma ← gamma × 2
 end

return best accuracy
end
```

Another classifier used in the experiments is Basic ELM. Basic ELM is used only for V1 and V2. Kernel functions used in the Basic ELM is sinus and sigmoid. Similar to SVM, hidden number of nodes parameter value of ELM is selected from a set of values. Hidden number of nodes parameter is started with the value which is equal to number of features in the dataset divided by 2, and at the end of each iteration it is increased by value equal to number of features in the dataset divided by 2 till reaching maximum value which is smaller than number of features in the dataset multiplied by 10. In the Figure 4.4, pseudocode of hidden number of node selection is given.

Figure 4.4: Basic ELM parameter selection

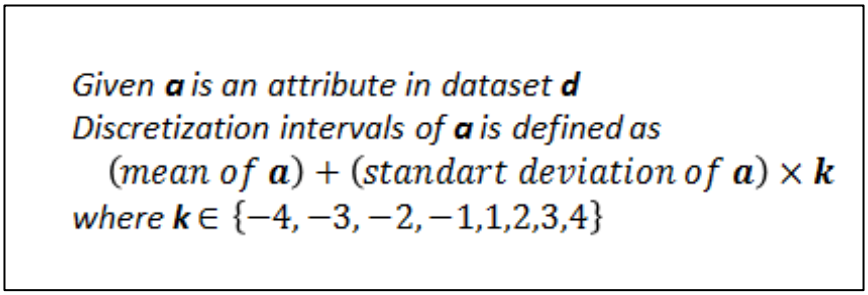
```
procedure elm(training,test)
accuracy ← 0
best accuracy ← 0
hidden node number ← number of attributes / 2
while hidden node number < number of attributes × 10
    accuracy ← classification(hidden node number,training,test)
    if best accuracy < accuracy then
      best accuracy = accuracy
    end
    hidden node number ← hidden node number + number of attributes / 2
end
return best accuracy
end
```

4.2 FEATURE SELECTION AND EXTRACTION

Both feature selection and extraction methods are applied to V1 because V2 has no significant dimensionality size. PCA algorithm implemented in WEKA is applied to V1 for feature extraction. The number of features extracted with PCA has been determined such that 93, 89, and 85 percent of the variance are preserved. The obtained set of features are fed to SVM and ELM classifiers in a supervised learning manner. Supervised learning classifiers' setup are the same as described in section 4.1.

In feature selection, mRMR is used. Before mRMR, using all samples, V1 data is discretized to 9 intervals where boundries of intervals are defined with formula in Figure 4.5 (Sakar et al. 2012, Peng et al. 2005).

Figure 4.5: Discretization of V1



After discretization, mRMR is applied by using its difference version, which is based on using the mutual information computed as the difference between relevance and redundancy terms, and the mRMR feature list is acquired. Then, the mRMR features are evaluated in an iterative classification process where the classification starts using all of the features and each time the last feature with minimum mRMR score on the list is removed, starting from 115th feature down to 1st feature in the list so that the minimal subset of features is selected. In the classification process, both Basic ELM and C-SVC SVM are used. In ELM, only sigmoid kernel is used with no parameter selection where value of number of hidden nodes is set to feature count multiplied by 5. Similarly, in SVM, only linear kernel is used with cost selection out of values 2, 16 and 128 according to test set accuracy. Classifier setups in feature selection process are less variate compared to supervised learning experiments due to computational performance reasons. Classifications are performed with 80 training samples per each class and rest of the samples are used for testing. All classifications are repeated 10 times and average accuracy is reported. In Figure 4.6, mRMR evaluation algorithm pseudocode is given.

Figure 4.6: mRMR features evaluation

```
procedure mRMR attribute evaluation(mRMR attribute list)
attribute list ← mRMR attribute list
 while size(attribute list) > 1
   for i ← 1 to 10
    training ← select x samples randomly for each class
    test ← rest of the samples
    accuracy ← accuracy + classifier(attribute list, training,test)
   next i
    accuracy ← accuracy / 10
    attribute list ← remove last attribute from attribute list
 end
end
```

According to the output of mRMR features evaluation, best performing feature set is selected and used in supervised learning with setup as described in section 4.1 with 80 training samples from each class to see if there is an improvement in the accuracy.

4.3 SEMI-SUPERVISED LEARNING

Unlike treating short-text views as seperate datasets which is done in supervised learning, all views of short-text instances are utilized together with an ensemble learning manner in semi-supervised learning. Each view has its own classifier which outputs its individual predictions for the test samples. For an unlabeled sample, these individual results of views are combined, and in case of consensus, unlabeled sample is labeled and incorporated to the training dataset to be used in the model construction of the next training iteration.

Dataset is splitted into 4 sets randomly as training, validation, test and unlabeled where there is equal sample distribution per class except in unlabeled set. Test and validation set size is fixed to 20 samples per class in all experiments while 20, 40 and 60 samples per class for initial training set are experimented seperately. For each view of initial

training samples, a seperate C-SVC type SVM with linear kernel is trained. Cost parameter value of linear kernel is choosen as described in supervised learning section (Figure 4.2) except instead of test set, validation set is used for parameter selection. Cost stars from 0.5 and multiplied by 2 in the end of each iteration till reaching 512 and best accuracy is choosen on validation set. For each view, cost parameter is choosen seperately since each view has its own svm classifier model and while using validation set respective view of validation sample is used. After selection of cost parameter value for each view, classification of unlabeled samples begin. An unlabeled sample is classified by each view's classifier according to its respective view. If all views of a sample are classified with the same class label, which is named as view consensus, then sample is choosen to be used in training data of the next iteration. When number of samples choosen with consesus reach 100 (maximum treshold) or all unlabeled samples are classified and there are more than 5 (minimum treshold) chosen samples, then chosen samples are added to initial training data and removed from the unlabeled sample set. At this point, all process starts again, classifiers of all views trained with new training set. Same cost parameter value selection method is applied as before and unlabeled samples are classified to select new samples to training set with consensus. This iterative ensemble learning process continues till the number of samples chosen with consensus is less than 5 or there are no unlabeled samples left which means all samples are labeled with consensus. In case consensus is below the minimum treshold, test set is used with classifiers and the accuricies of each view classifier is obtained. All experiment is done for 10 times with random selection of training, validation, and test set each time, and average accuracy is reported. In Figure 4.7, the pseudo-code of the semi-supervised learning scheme used is given.

Figure 4.7: Semi-supervised learning

```
procedure semi-supervised learning(x,y,z)
accuracy ← 0
for i ← 1 to 10
    training ← select x samples randomly from each class
    validaton ← select y samples randomly from each class in remaining set
    test ← select z samples randomly from each class in remaining set
    unlabeled ← rest of the samples in remaining set
    while size(unlabeled) > 0
        view classifier models ← svm linear(training, validation)
        consensus ← ensemble learning(unlabeled, view classifier models)
        if size(consensus) < 5 then
            break while loop
        end
        training ← training + consensus
        unlabeled ← unlabeled - consensus
    end
    accuracy ← accuracy + svm linear(test, view classifier models)
 next i
accuracy ← accuracy / 10
end
```

5. RESULTS

In this section, the experimental results obtained with the methods described in Section 4 are given. The results are given in three subsections which are Supervised Learning Results, Feature Selection and Extraction Results, and Semi-Supervised Learning Results. Each experiment is repeated 10 times for statistical significance with a randomly selected set of samples each time and mean accuracy is given.

5.1 SUPERVISED LEARNING RESULTS

Training set sizes given in the tables show the number of samples used for training from each class. Rest of the samples are used for testing. In Tables 5.1, 5.2, and 5.3 the supervised learning results are given for V1, V2, and V3, respectively.

Table 5.1: V1 supervised learning results

V1 Sample Count per Class	**SVM**		**ELM**	
	linear	***RBF***	***sig***	***sin***
80	73.164	**73.183**	47.148	37.636
60	**72.468**	71.123	44.762	38.101
40	**69.401**	68.723	40.452	37.18
20	**63.061**	61.318	36.766	35.974

Table 5.2: V2 supervised learning results

V2 Sample Count per Class	SVM		ELM	
	linear	***RBF***	***sig***	***sin***
80	69.98	**70.195**	63.242	70.898
60	**69.351**	68.829	63.512	61.55
40	**68.843**	66.715	62.526	56.289
20	**64.839**	62.041	55.699	42.19

Table 5.3: V3 supervised learning results

V3 Sample Count per Class	SVM	
	linear	***RBF***
80	86.777	**88.73**
60	84.683	**86.344**
40	81.223	**82.154**
20	71.215	**71.594**

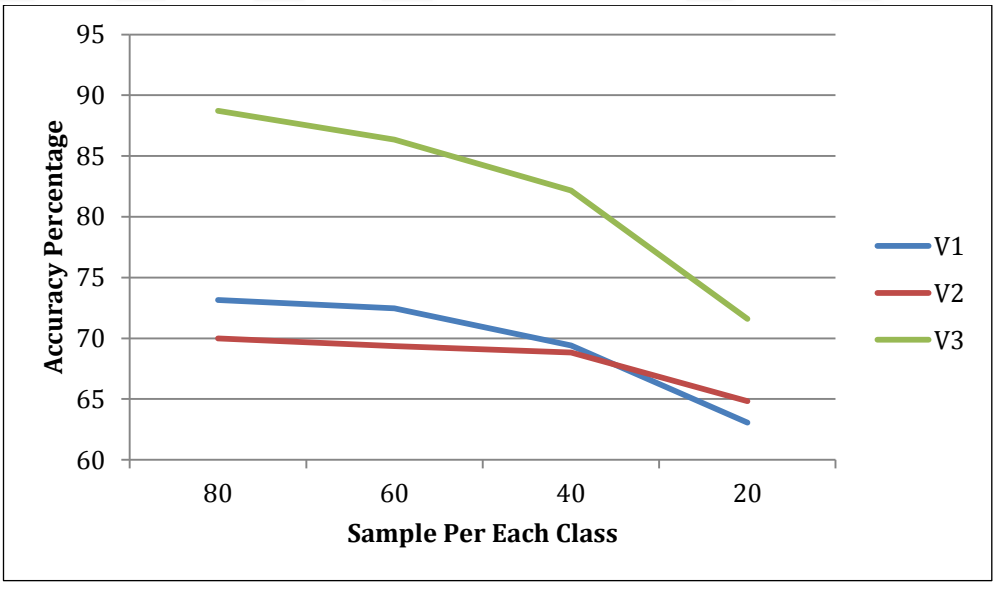
Considering all the training set sizes, SVM with linear kernel gave higher accuracies than ELM for V1 and V2. It should be noted that accuracies obtained with ELM on V1 are too low to be used in a real application. Although ELM and SVM are comparable on V2 with 80 samples, overall evaluation shows that SVM performs significantly better than ELM on the supervised short-text classification problem. For V3, RBF kernel is slightly better than linear kernel. In Table 5.4, best accuracies of views are compared.

Table 5.4: Best supervised learning results compared

Sample number per class	V1 (SVM Linear)	V2 (SVM Linear)	V3 (SVM RBF)
80	73.164	69.98	88.73
60	72.468	69.351	86.344
40	69.401	68.843	82.154
20	63.061	64.839	71.594

Considering all the training set sizes, V3 representation seems more successful than V1 and V2. V1 is better than V2 except the case 20 samples per class are used for training. Another important result is that the accuracy of V2 decreases by 5% when the number of training samples used for each class is reduced from 80 to 20 whereas the accuracies of V1 and V3 decrease by 10% and 17%, respectively. In Figure 5.1, the accuracies of the best performing supervised learning algorithms are shown with respect to the number of training samples selected from each class.

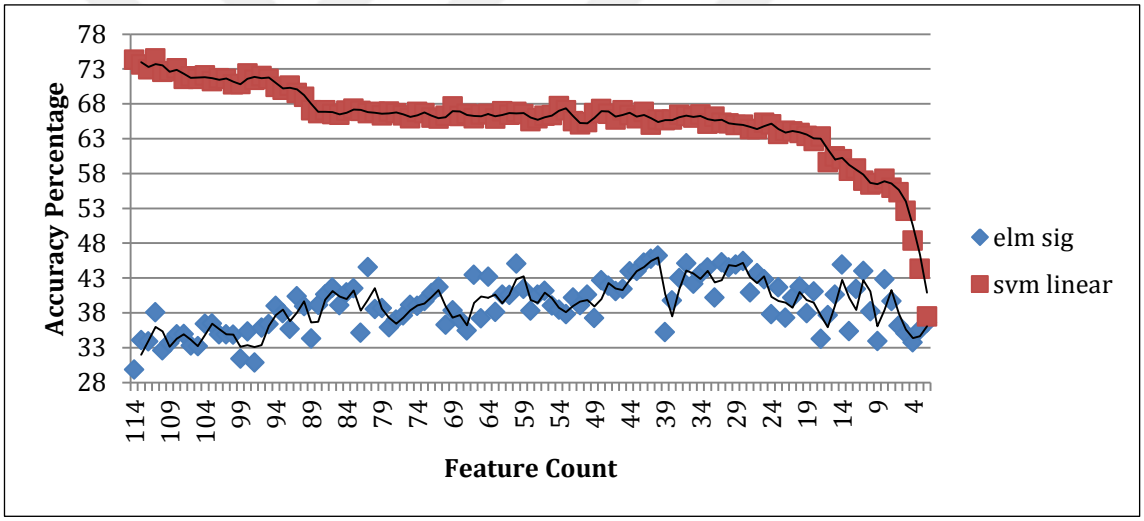
Figure 5.1: Best supervised learning results graph



5.2 FEATURE SELECTION AND EXTRACTION RESULTS

Feature selection and extraction is done for V1 dataset. In feature selection, mRMR provided with input of all V1 dataset and MID option is used. mRMR features list, which is the output of mRMR, are used in an iterative supervised classification process where the classification starts using all of the features and each time the last feature with minimum mRMR score in the list is removed from V1. For supervised learning 80 samples per each class is randomly taken for training and rest is used for testing. Each supervised classification is executed for 10 times with random training and test samples and mean accuracy is given. In Figure 5.2, mRMR feature selection graph is given.

Figure 5.2: mRMR feature selection graph



Feature selection with mRMR has no positive effect on SVM while for ELM it has an increasing trend. SVM and ELM classifiers in feature selection do not have the same setup with supervised learning SVM and ELM setups due to computational performance concerns. Due to this reason, according to the output of feature selection in ELM, 29 first features are selected where it is in the middle of the increase trend, and then experimented with supervised learning setup ELM which the results are given in Table 5.5. Also in Table 5.5, application of SVM with supervised learning setup on PCA output for 93, 89 and 85 percent variance coverages are given. For experiments in Table

5.5, training is composed of randomly selected 80 samples per each class and rest of the samples is used for testing. Each supervised classification is executed for 10 times with random selection of training and test samples and mean accuracy is given.

Table 5.5: Feature selection and extraction accuracies

V1 80 sample per class	SVM		ELM	
	linear	***RBF***	***sig***	***sin***
original	73.164	73.183	47.148	37.636
PCA 93 variance	67.226	60.527	50.957	25.214
PCA 89 variance	66.738	61.23	48.378	24.296
PCA 85 variance	66.25	60.625	50.214	26.699
mRMR (first 29 for ELM)	-	-	58.984	67.246

Feature extraction with PCA has no positive effect on accuracy for V1 dataset. Although mRMR feature selected version of V1 has an increase in accuracy compared to original version in ELM, especially 30 percent in sinus kernel, SVM with original V1 dataset is still the most successful clearly.

5.3 SEMI-SUPERVISED LEARNING RESULTS

In semi-supervised learning 60, 40 and 20 randomly selected samples per each class is experimented as **starting** training set while for all experiments 20 randomly selected samples for validation per each class, 20 randomly selected samples for test per each class and remaining samples as unlabeled set are used. At the end of ensemble learning process, where there are not enough unlabeled samples to be learned with consensus of three views, classifiers are tested on test set and accuracy captured. All semi-supervised learning process is executed for 10 times with random data distribution and mean of accuracies are captured. In Table 5.6, results of semi-supervised learning process are given together with best recorded supervised learning results.

Table 5.6: Semi-supervised versus supervised learning

Sample number per class	V1		V2		V3	
	Semi Sup.	Sup.	Semi Sup.	Sup.	Semi Sup.	Sup.
60	70.58	72.468	66.50	69.351	85.67	86.344
40	70.08	69.401	68.00	68.843	85.58	82.154
20	67.33	63.061	66.08	64.839	74.58	71.594

As the starting training number decreases from 60 to 40 sample per each class, there is no significant change in the accuracy of semi-supervised learning. But from 40 to 20, decrease in accuracy seen. As it was in supervised learning, the least affected view from sample size decrease is V2 while the most affected is V3 in terms of accuracy.

Supervised learning is better compared to semi-supervised learning at the 60 starting training number per each class, with 2 percent, 3 percent and 1 percent for V1, V2 and V3 respectively. At 40 starting training number per each class, V1 and V2, semi-supervised and supervised accuracies are balanced while semi-supervised learning is 3 percent better for V3. At 20 starting training number per each class, semi-supervised learning is better than supervised learning for all views with 4 percent, 1 percent and 3 percent for V1, V2 and V3 respectively. Accuracy difference for V3 in 20 and 40 samples per each class is not changed. In figures 5.3, 5.4 and 5.5, semi-supervised versus supervised learning accuracy graphs are given for V1, V2 and V3 respectively.

Figure 5.3: V1 semi-supervised versus supervised

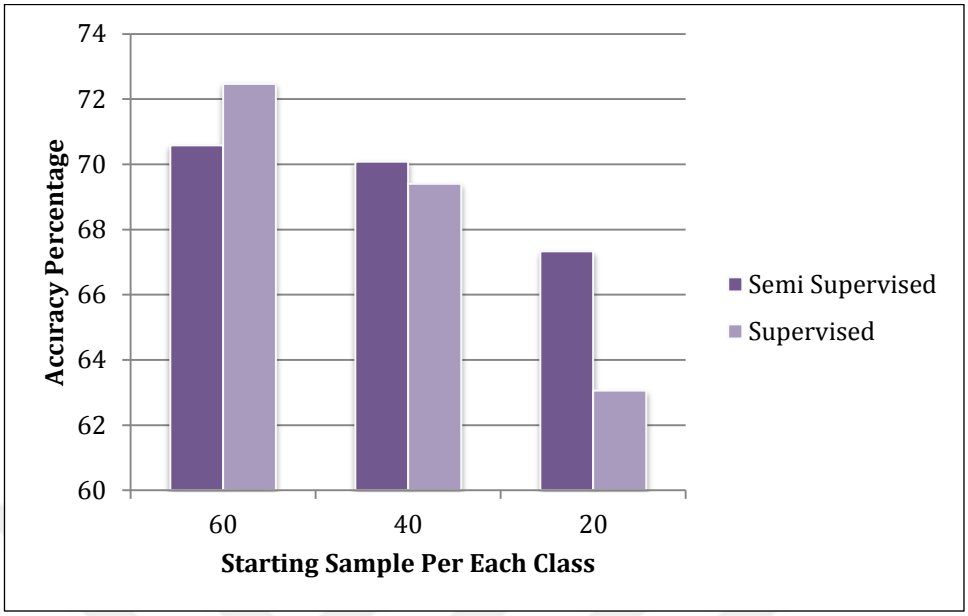


Figure 5.4: V2 semi-supervised versus supervised

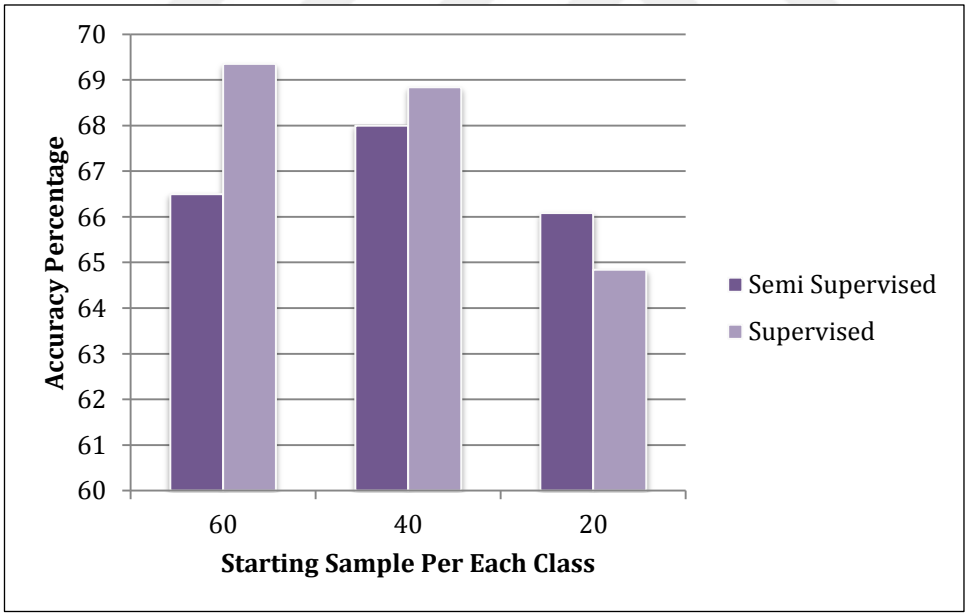
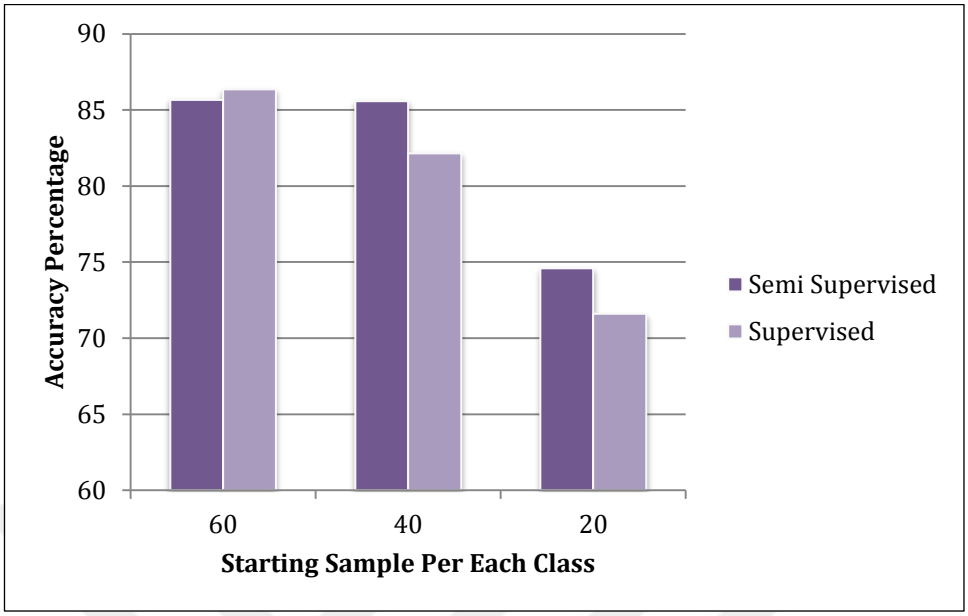


Figure 5.5: V3 semi-supervised versus supervised



6. DISCUSSION

It is important to note that results of the generated views using knowledge base are highly dependent on the quality of the data that forms the knowledge base which in this thesis it is Turkish Wikipedia (Vikipedi). As Poyraz et al. (2012) states, Turkish Wikipedia contains quite a lot of noise. Besides, half of the generated 3-gram features in V1 have no value which is a loss of semantic information. This signals that there can be a coverage problem of Turkish Wikipedia's domain specific features which are utilized in this thesis, although it depends on the extracted set of articles, their topics and used seed short-text dataset. For this reason, results must be evaluated considering these limitations.

The experimental results show that, views generated with proposed features using proposed knowledge base cannot be an alternative to the BoW representation for short-text classification. However, knowledge base generated views are more resistant to accuracy loss compared to BoW when sample size decrease in training data. This originates from the fact that the dimensionality of the knowledge base generated views is far less than that of BoW representation, and the classifiers can generalize better with lower input dimensionality. This makes the generated views useful for semi-supervised learning where the number of labeled samples is small compared to the number of unlabeled samples.

The results obtained using a filter feature selection algorithm show that, as expected, SVM is robust to input dimensionality. It has been observed that ELM is more sensitive to input dimensionality than SVM. Using mRMR for feature selection as a preprocessing step improves the accuracy of ELM on V1 while it decreases the accuracy of SVM. Although mRMR shows that ELM is affected from curse of dimensionality, PCA application on V1 dataset decreases the accuracy for ELM in sinus kernel and has a little change on sigmoid kernel compared to mRMR. This indicates that

features in V1 dataset are strong features. Also for SVM, application of PCA decreases the accuracy.

For used methods in this thesis in short-text classification, semantic features generated from Wikipedia domain specific properties have no significant success over features generated from Wikipedia article plain text. This shows that external high quality plain text data without any specific properties can also be a good source of knowledge base. Results show that ensemble learning with multiple views in semi-supervised learning of short-texts increase the accuracy of every view only when the number of samples in training set is very small.

7. CONCLUSION

In this thesis, short-text classification problem in Turkish language with limited number of labeled training samples and a large amount of unlabeled samples is addressed. A popular method, using an external data source as knowledge base, is implemented in order to overcome the sparseness and lack of semantic information problems of Bag-of-Words (BoW) representation. The knowledge base is constructed from encyclopedic knowledge due to its effectiveness in short samples.

Multiple views of short-text samples are generated from the original samples. The first and second views are the representations of short-texts in knowledge base while the third view is traditional BoW. Turkish Wikipedia (Vikipedi) articles are extracted, parsed, stored, and used as the knowledge base. Some of the existing methods that use external knowledge bases are not suitable for real time tasks and scenarios due to their high computational complexity. Hence, in this thesis, knowledge base storage is done in a way that enables quick referencing for generation of samples' views.

The features of the first view are built on top of Wikipedia's domain specific properties like titles and anchor texts to preserve the semantic information. The features of the second view are built on top of article plain texts which can also be built on top of another external source instead of Wikipedia. Both views contain n-gram features for adding semantic information and both views do not contain any feature terms like BoW. As a result, two views which are generated using knowledge base possess semantic information. Besides, the sparseness problem of BoW had overcome.

The traditional BoW and knowledge base generated views are compared in terms of their success when used as input in supervised learning problem with different number of training sample sizes. The results show that knowledge base generated views can be used together with the BoW representation in an ensemble manner in order to increase the accuracy and generalization of the classifier especially when there are not sufficient

number of training samples. Also it is observed that features generated from Wikipedia domain specific properties which are expected to add additional semantic value have no significant difference on accuracy compared to features generated from article plain text.

All views of short-text sample is used together for ensemble learning in semi-supervised classification, and the results show that this approach increases the accuracy for small number of training samples.

As the results are highly dependent upon the quality of the source of knowledge base, which is Turkish Wikipedia in this thesis, it can be concluded that the final accuracy obtained with our model may be further improved by increasing the quality and coverage of the knowledge base.

REFERENCES

Books

Gabrilovich, E. and Markovitch, S., 2006. Overcoming the brittleness bottleneck using Wikipedia: enhancing text categorization with encyclopedic knowledge. *National Conference on Artificial Intelligence (AAAI)*, AAAI Press, pp. 1301-1306

Huang, A., Milne, D., Frank, E., Witten, I.H., 2008. Clustering document with active learning using Wikipedia. *Eighth IEEE International Conference on Data Mining, 2008. ICDM'08.* IEEE, pp. 839 - 844

Joachims, T., 1998, Text categorization with support vector machines: learning with many relevant features. *ECML'98 Proceedings of the 10th European Conference on Machine Learning*. Verlag London: Springer, pp. 137 - 142

Matsubara, E.T., Monard, M.C. and Batista, G.E.A.P.A, 2005, Multi-view semi-supervised learning: an approach to obtain different views from text datasets. *Proceedings of the 2005 conference on Advances in Logic Based Intelligent Systems: Selected Papers of LAPTEC 2005*. Amsterdam Netherlands: IOS Press, pp. 97 - 104

Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H. and Demirbas, M., 2010, Short-text classification in twitter to improve information filtering. *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. New York NY USA: ACM, pp. 841 – 842

Zhang, C., Xue, G.R. and Yu, Y., 2008. Knowledge supervised text classification with no labeled documents. *PRICAI 2008:Trends in Artificial Intelligence, 10th Pacific Rim International Conference on Artificial Intelligence, Hanoi, Vietnam, December 15-19, 2008. Proceeding.* Berlin Heidelberg: Springer, pp. 509-520.

Zhang, Z., Lin, H., Li, P. Wang, H. and Lu, D., 2013. Improving semi-supervised text classification by using Wikipedia knowledge. *Web-Age Information Management, 14th International Conference, WAIM 2013, Beidaihe, China, June 14-16, 2013. Proceedings.* Berlin Heidelberg: Springer, pp. 25-36.

Periodicals

Chang, C.C. and Lin, C. J., 2011, A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*. **2**(3), pp. 1 – 27

Cortes, C. and Vapnik, V., 1995, Support-vector networks. *Machine Learning*. **20**(3), pp. 273 - 297

Ding, S., Zhao, H., Zhang, Y., Xu, X. and Nie, Ru., 2015, Extreme learning machine: algorithm, theory and applications. *Artificial Intelligence Review*. **44**(1), pp. 103 – 115

Gu, P., Zhu, Q. and Zhang, C., 2009, A multi-view approach to semi-supervised document classification with incremental naïve bayes. *Computers & Mathematics with Applications*. **57**(6), pp. 1030 - 1036

Gupta. V and Lehal, G. S., 2009, A survey of text mining techniques and applications. *Journal of Emerging Technologies in Web Intelligence*. **1**(1), pp. 60 – 76

Guran, A., Bayazit, N. G. and Gurbuz, M. Z., 2013, Efficient feature integration with Wikipedia-based semantic feature extraction for Turkish text summarization. *Turkish Journal of Electrical Engineering and Computer Sciences*. **21**(5), pp. 1411 – 1425

Hall, M., Eibe, F., Holmes, G., Pfahringer, B. and Reutemann P., 2009, The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*. **11**(1), pp. 10-18.

Huang, G.B., Zhu, Q.Y. and Siew, C.K., 2006, Extreme learning machine: theory and applications. *Neurocomputing*. **70**(1-3), pp 489 - 501

Li, G., Chang, K. and Hoi, S.C.H., 2012, Multiview semi-supervised learning with consensus. *IEEE Transactions on Knowledge and Data Engineering*. **24**(11), pp 2040 - 2051

Man, Y., 2014, Feature extension for short-text categorization using frequent term sets. *Procedia Computer Science*. **31**(2014), pp. 663 – 670

Peng, H., Long, F. and Ding, C., 2005, Feature selection based on mutual information: Criteria of max-dependency, max relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* **27**(8), pp. 1226 – 1238

Sakar, C.O., Kursun, O. and Gurgen, F., 2012, A feature selection method based on kernel canonical correlation analysis and the minimum Redundancy-Maximum Relevance filter method. *Expert Systems with Applications*. **39**(3), pp. 3432 - 3437

Sebastiani, F., 2002, Machine learning in automated text categorization. *ACM Computing Surveys*. **34**(1), pp. 1 - 47

Wang, B. K., Huang, Y. F., Yang, W. X. and Li, X., 2012, Short-text classification based on strong feature thesaurus. *Journal of Zhejiang University SCIENCE C*. **13**(9), pp. 649 – 659

Other Publications

Basic Extreme Learning Machine (ELM) Implementation in Java. [online] http://www3.ntu.edu.sg/home/egbhuang/source_codes/ELM-Java.zip. [accessed 01 September 2015]

Poyraz, M., Ganiz, M. C., Akyokuş, S., Görener B. and Kilimci, Z. H., 2012, Exploiting Turkish Wikipedia as a semantic resource for text classification. *Innovations in Intelligent Systems and Applications (INISTA), 2012 International Symposium, Trabzon*. IEEE, pp. 1 - 5

Rafi, M., Hassan, S. and Shaikh, M. S., 2012, Content-based text categorization using Wikitology. *Computing Research Repository.* **abs/1208.3623**

TS Abstract Corpus. [online] http://tanersezer.com/?p=203. [accessed 23 August 2015].

Wikipedia. [online] https://tr.wikipedia.org. [accessed 23 August 2015].

Wikipedia CatScan version2. [online] https://tools.wmflabs.org/catscan2/catscan2.php. [accessed 23 August 2015].

Wikipedia Export page for Turkish Wikipedia (Vikipedi). [online] https://tr.wikipedia.org/wiki/%C3%96zel:D%C4%B1%C5%9FaAktar. [accessed 23 August 2015].

Yıldırım, O., Atık, F., 2013. Senior Project, Department of Computer Engineering, Yıldız Technical University. [online] http://www.kemik.yildiz.edu.tr/data/File/42bin_haber.rar. [accessed 10 October 2015].

Zemberek. [online] https://github.com/ahmetaa/zemberek-nlp. [accessed 01 March 2015].