Mühendislik Bilimleri ve Tasarım Dergisi
9(4), 1048 – 1061, 2021
e-ISSN: 1308-6693

Journal of Engineering Sciences and Design
DOI: 10.21923/jesd.780002

Araştırma Makalesi Research Article

CLUDS: COMBINING LABELED AND UNLABELED DATA WITH LOGISTIC REGRESSION FOR SOCIAL MEDIA ANALYSIS

Berna ALTINEL GIRGIN *

Marmara University, Faculty of Technology, Dept. of Computer Engineering, Istanbul, Turkey

Keywords

Tweet Classification, Latent Dirichlet Allocation, Logistic Regression, Social Media Analysis, Sentiment Polarity Detection.

Abstract

Automatic text classification and sentiment polarity detection are two important research problems of social media analysis. The meanings of the words are so important that they need to be captured by a document classification algorithm to reach an accurate classification performance. Another important issue with the text classification is the scarcity of labeled data. In this study, Combining Labeled and Unlabeled Data with Semantic Values of Terms (CLUDS) is presented. CLUDS has the following steps: preprocessing, instance labeling, combining labeled and unlabeled data, and prediction. In preprocessing step Latent Dirichlet Allocation (LDA) algorithm is used. In instance labeling step Logistic Regression is applied. In CLUDS, relevance values computation has been applied as a supervised term weighting methodology in the text classification field. Still, according to the literature, CLUDS is the first attempt that uses both relevance and weighting calculation in a semi-supervised semantic kernel for Support Vector Machines (SVM). In this study, Sprinkled-CLUDS and Adaptive-Sprinkled-CLUDS have also been implemented. Evaluated experimental results show that CLUDS, Sprinkled-CLUDS and Adaptive-Sprinkled-CLUDS generate a valuable performance gain over the baseline algorithms on test sets.

CLUDS: SOSYAL MEDYA ANALİZİ İÇİN ETİKETLİ VE ETİKETSİZ VERİLERİ LOJİSTİK REGRESYON İLE BİRLEŞTİRME

Anahtar Kelimeler

Tweet Sınıflandırması, Gizli Dirichlet Analizi, Lojistik Regresyon, Sosyal Medya Analizi, Duygu Polarite Tespiti.

Öz

Otomatik metin sınıflandırması ve duygu polarite tespiti, sosyal medya analizinin iki önemli araştırma problemidir. Kelimelerin anlamları o kadar önemlidir ki, doğru bir sınıflandırma performansına ulaşmak için bir belge sınıflandırma algoritması tarafından yakalanmaları gerekir. Metin sınıflandırmasıyla ilgili bir diğer önemli konu, etiketlenmiş verilerin azlığıdır. Bu çalışmada, yeni bir yarı denetimli metodoloji sunulmuştur. Etiketli ve Etiketlenmemiş Verilerin Anlamsal Terim Değerleri (CLUDS) ile Birleştirilmesi olarak adlandırılır. CLUDS şu adımlara sahiptir: ön işleme, örnek etiketleme, etiketli ve etiketlenmemiş verileri birleştirme ve tahmin. Ön işleme adımında Latent Dirichlet Allocation (LDA) algoritması kullanılmaktadır. Örnek etiketleme adımında Lojistik Regresyon uygulanır. CLUDS'ta, alaka değerleri hesaplaması, metin sınıflandırma alanında denetimli bir terim ağırlıklandırma yöntemi olarak uygulanmıştır. Literatüre göre, CLUDS, Destek Vektör Makineleri (SVM) için yarı denetimli bir semantik çekirdekte hem alaka düzeyi hem de ağırlık hesaplamasını kullanan ilk girişimdir. Bu çalışmada, Sprinkled-CLUDS ve Adaptive-Sprinkled-CLUDS da uygulanmıştır. Değerlendirilen deney sonuçları CLUDS, Sprinkled-CLUDS ve Adaptive-Sprinkled-CLUDS'ın test setlerinde temel algoritmalara göre değerli bir performans kazancı sağladığını göstermektedir.

Alıntı / Cite

Altınel Girgin, B., (2021). CLUDS: Combining Labeled and Unlabeled Data With Logistic Regression For Social Media Analysis, Journal of Engineering Sciences and Design, 9(4), 1048-1061.

* İlgili yazar / Corresponding author: berna.altinel@marmara.edu.tr, +90-216-348-0292

Yazar Kimliği / Author ID (ORCID Number)	Makale Süreci / Article Process	
B. Altınel Girgin, 0000-0001-5544-0925	**Başvuru Tarihi / Submission Date**	13.08.2020
	Revizyon Tarihi / Revision Date	18.08.2021
	Kabul Tarihi / Accepted Date	06.09.2021
	Yayım Tarihi / Published Date	20.12.2021

1. Introduction

1.1 Conventional Text Classification

Conventional text classification approach and semantic text classification approach can be compared along many criteria. These criteria are data representation, features, ontology, background source, functional focus, synonymy, polysemy, ambiguity, analysis, applicability in social networks and classification accuracy. The semantic text classification approach has many benefits over the traditional approach. In semantic text classification, both the semantic connections between the words and the documents are considered. In the semantic approach, a background source can be used to get extra information about the terms such as polysemy, synonymy, ambiguity etc. Moreover, statistical computations from the corpus in the corpus-based systems mine the hidden connections between the terms and the documents, which advance the classification accuracy. There is an extensive literature survey about the comparison of the traditional text classification approach and semantic text classification approach in (Altınel & Ganiz, 2018).

Semantic methods can be classified into five groups (Altınel & Ganiz, 2018). domain knowledge-based (ontology-based) approaches (Bloehdorn & Moschitti, 2007; Fung, 2003; Moore, 2003; Muslea et al., 2002; Nigam et al., 2000; Papka &, Allan, 1998; Reborto, 2012), corpus-based approaches (Altınel et al., 2015; Bai et al., 2004; Liu et al., 2004; Uysal & Gunal, 2014; Zhou et al., 2008), word sequence enhanced approaches (Fung, 2003; Peng et al., 2003; Razon & Barnden, 2015), linguistic enriched approaches (Nigam et al., 2000) and deep learning based approaches (Bengio et al., 2008; Bordes et al., 2012; Dahl et al., 2010; Dahl et al., 2012; Hinton et al., 2012; Hinton et al., 2006; Krizhevsky et al., 2012; Mikolov et al., 2011; Seide et al., 2011).

There are three traditional approaches in machine learning applications supervised learning, unsupervised learning and semi supervised learning (SSL). Conventional supervised learning approaches require a number of training documents with their class labels to generate the classifier that will then be used to predict the class labels of the test instances. Conversely, unsupervised learning merely depends on unlabeled examples. Thus, unsupervised learning tries to find out the latent patterns of unlabeled data to train a classifier (Zhu, 2005). Inopportunely most of the data on the web do not have class labels, which restrict their use in numerous application fields like sentiment recognition, text classification and speech recognition. In addition, assigning labels to unlabeled documents manually can be tedious and expensive. Most prominently, building a classifier model by using only a small number of labeled data may not create satisfactory classification accuracy. In circumstances where labeled data are insufficient, several methodologies have been proposed which use the unlabeled instances. Different from these two approaches, the SSL approach utilize both labeled data and unlabeled samples to increase the classification performance. There are different kinds of SSL algorithms that have been proposed in the literature in previous years: transductive SVM (Chapelle et al., 2005), Estimation-Maximization (EM) with generative mixture models (Nigam et al., 2000a; Nigam et al., 2000b), graph-based algorithms (Zhu, 2005), self-training (Rosenberg et al., 2005; Yarowsky, 1995) and co-training (Blum & Mitchell, 1998). Self-training and co-training are two widely used algorithms among them.

1.2 Social Media Analysis

Social Media is actually like a revolution in human's life since it completely changed people's communication, shopping, working…etc. styles. After the spread of the Internet, many things in people's daily life have started to able to be done just with a computer or a cellular phone without not going to anywhere. This, of course, contributed to the human's life very much simplicity. On the other hand, these new trends result very huge amounts of accumulated data on the communication channels, microblogging sites, review forms…etc. This actually increases the importance of automatically categorization of textual materials on WWW.

There are very important research problems related to Social Media like: 1.) Spam filtering (Ferrara et al., 2014; Hu et al., 2014a, 2014b; Yardi et al., 2009), 2.) Opinion Leader Detection (Amor et al., 2016; Luo et al., 2018; Zhao et al., 2015), 3.) Information Diffusion (Cho et al., 2012; Kempe et al., 2003; Van et al., 2011; Zhao et al., 2016a), 4.) Short Text Classification (Wang et al., 2017; Zeng et al., 2018), 5.) Sentiment Analysis (Denecke, 2008; Khan et al., 2016; Pang et al., 2002)…etc. Several types of Un/Semi/Supervised algorithms from both Machine Learning and Deep Learning are used on those studies (Injadat et al., 2016).

1.3 Short Text Classification

Since commonly, used web sites are like microblogging sites, question-answering channels, news headlines, forum messages, and status updates; so the accumulated data on the web are generally in the form of short text. This results very large volume of short texts available on the Internet. In order to classify short textual materials several methodologies have been presented such as, naive Bayes (Wang & Manning, 2012), using SVMs with rule-based features (Silva et al., 2011) and constructing dependency trees with Conditional Random Fields (Nakagawa et al., 2010), convolutional neural networks (CNNs) (Kalchbrenner et al., 2014) and recursive neural networks (Kalchbrenner et al., 2014).

1.4 Sentiment Polarity Detection

In addition to topic classification, sentiment polarity detection is also another important task. People have many opportunities to write their comments, feelings, and reviews on social media. Sentiment polarity detection is the task of classifying reviews/comment/short texts into positive and negative category based on the total semantic polarity they carry on. Especially, on microblogs, review-pages, sentiment classification would also be helpful for recommender systems (Salah et al., 2019; Mishne & Glance, 2006), spam filtering (Hu et al., 2014; Peng & Zhong, 2014) and decision making (Chalothom & Ellman, 2015; Koehler et al., 2015).

In a study (Asiaee et al., 2012) authors collect tweets under some predefined categories and them classify them according to their sentiment polarity. They present a cascaded classifier model with three sequential 2-class classification steps. In their cascaded model, there are two main steps: 1.) isolating tweets that are about the topic of interest, 2.) removing tweets that do not contain any emotion. In order to compare the classification performance of their algorithm they used several baseline algorithms such as NB, SVM, and KNN. They conduct three datasets including 4490, 8850, and 12770 tweets in their experiment environment. According to their experimental results, they achieve better classification accuracy in compare to the baseline algorithms.
Combining Labeled and Unlabeled Data with Semantic Values of Terms (CLUDS):

Recently, an original supervised semantic kernel for SVM (Altınel et al., 2015) is presented: Class Weighting Kernel (CWK). CWK takes advantages of weighting calculation, which is based on (Biricik et al., 2009, 2012). This weighting calculation is used to generate a semantic matrix, which indicates the semantic closeness between words. The experimental results show that CWK has an important gain in classification accuracy over linear kernel (Altınel et al., 2015).

In this study, a new semi-supervised methodology is presented, which is named Combining Labeled and Unlabeled Data with Semantic Values of Terms (CLUDS). This is a hybrid algorithm, which has two different semantic kernels with different aims. CLUDS utilizes both labeled and unlabeled data for building a classification model. Firstly, it generates a proximity matrix, which includes the relevance values (Lan et al., 2009) of words in the training corpus and directly indicates the semantic relationships between words. Then, it tries to classify unlabeled samples with a new classification algorithm, namely, Relevance Values Kernel (RVK) and moves them into the originally labeled set. In the second part of the algorithm, another semantic kernel, CWK (Altınel et al., 2015) is applied. The main novelty of CLUDS is the application of class-based relevance and weighting calculations in classification step. It is observed that CLUDS generates noticeable classification gains over the baseline algorithms in the experiments.

One of the most important advantages of CLUDS is its classification gain over the baseline algorithms in the experiments. There are four different baseline algorithms in the experiments. CLUDS is superior to all of them in most of the labeled set percentages. Moreover, the superiority of CLUDS is mostly visible at low-labeled set levels, which are very important since it is very hard to find labeled instances in real life scenarios.

Another central benefit of CLUDS is that it does not require heavy use of the lexical databases and grammatical tags in order to extract semantic features and calculate the similarity between documents like knowledge-based systems. Additionally, an important restriction of using knowledge bases is that their scope and coverage can restrict the ability of methods (Yarowsky, 1995). Moreover, these kinds of resources are usually expensive to maintain and often not available for particular fields.

Sprinkled-CLUDS and Adaptive-Sprinkled-CLUDS have also been implemented. Sprinkled-CLUDS has the same architecture with CLUDS with the only difference that Sprinkled-CLUDS uses additional terms, which represent the class relationships between documents. In other words, class labels are added into standard term-document matrix in order to enrich the class knowledge in training corpus and add this information into classification model.

This paper is organized as following: Section 2 presents short text classification, weighting and relevance

calculations, sprinkling, Latent Dirichlet Allocation and Logistic Regression. Details of the suggested methodology, CLUDS, are given in Section 3. Experiment setup is given in Section 4. Experimental results are reported in Section 5. Finally, the conclusion is given in Section 6.

2. Literature Survey

2.1 Short Text Classification

Recently, in (Ahmed et al., 2015), the authors presented a frequent item set and ensemble learning based semi-supervised methodology to classify SMS with small amount of labeled SMS and huge number of SMS. According to their experimental results, their proposed method generates good results. Nevertheless, as they mention in (Ahmed et al., 2015), the success of their methodology closely depends on the dataset volume.

In (Shinnou et al., 2015), a novel algorithm is presented for the domain adaption problems of document classification. In their approach, Self-Training Feature Weight (Chen et al., 2009) is used to reconstruct training data, and then the final classification model is learned by using the reconstructed training instances and Naive Bayes. According to their experiments on 20 Newsgroups, the influence of their method is shown. In (Song et al., 2014), recent semi-supervised learning methodologies are mentioned and analyzed in detail.

In a recent work (Asiaee et al., 2012); a very simple and novel supervised term weighting approach is presented. In order to show the superiority of their proposed algorithm they conduct two datasets from Twitter and apply some machine learning algorithms. According to the experimental results, they report, their proposed term weighting approach, SW, result higher classification performance in compare to SVM, Decision Tree, k-nearest neighbor, and logistic regression algorithms. For instance, in (Alsmadi & Hoon, 2019) SW is reported to improve tweet classification accuracy to 80.83 and 90.64 (F-measure) on their self-collected Twitter dataset.

In another recent and interesting study (Hu et al., 2018); a new algorithm is proposed namely SVMCNN. SVMCNN combines Convolutional Neural Networks and Support Vector Machine. They conduct their own collected Twitter data for dataset. In SVMCNN model, CNN is used to extract features of short texts. After that, these features are used in SVM classifier for classification. According to the experimental results, reported in (Hu et al., 2018) SVMCNN model can achieve has good performance in short text classification.

2.2 Relevance Values of Terms

Term Frequency-Relative Frequency (TF-RF) is a term-weighting technique (Lan et al., 2009). The formula of TF-RF is:

$$\mathrm{TF-RF} = tf_w \times log\left(2 + \frac{a}{\max(1, c)}\right), \qquad (1)$$

where tfw presents the term frequency of word w, a shows the number of documents in the positive category, which include word, w and c denotes the number of documents in the negative category, which include word w. According to experimental results, RF methodology seems to give weights that are more appropriate in compare to traditional Inverse Document Frequency (IDF) technique (Lan et al., 2009). Table 1 reports the IDF and RF scores of four terms based on 00_acq and 03_earn categories. The scores listed in Table 1 (Lan et al., 2009) show that RF technique gives values that are more appropriate in compare to traditional IDF technique since RF uses class information, which improves the values.

2.3 Sprinkling

Sprinkling is a recent technique, which just adds additional terms corresponding to class labels of documents to the corpus that consists of training documents (Chakraborti et al., 2006). This process strengthens class-based relationships in the training phase. The sprinkling process is shown in Figure 1. There is an original term-document matrix with r documents and j terms in Figure 1(a). Sprinkled term-document matrix after sprinkling process with n terms is represented in Figure 1(b).

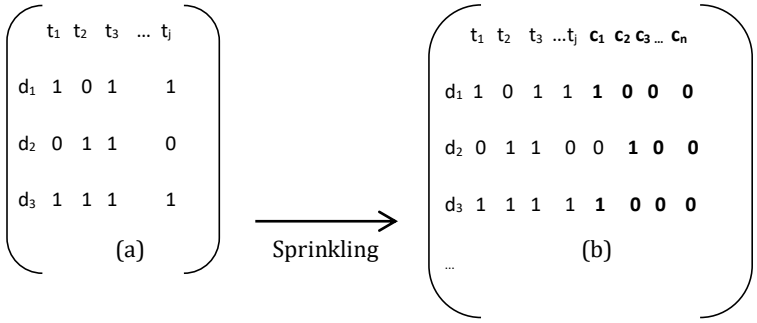

Figure 1. Sprinkling Process

The sprinkled term-document matrix includes additional terms show the class labels of the corresponding documents. For instance, d1 belongs to class c1, d2 belongs to class c2, d3 belongs to class c1, di belongs to class c2…., etc. These further features in training set directly contribute class information into the classifier.

2.4 Latent Dirichlet Allocation (LDA)

In CLUDS, for topic modeling, Latent Dirichlet Allocation (LDA) was used by applying the pooling method, in which the tweets they sent each day for each user were combined. LDA is a probabilistic topic modeling method, which generates a set of words and their corresponding weights for a predefined number of topic-labels of the corpus. In LDA method, topics have a probability distribution on words, and text documents have a probability distribution on topics. Each topic has a distribution on a fixed word sequence (Blei et al., 2003). The model aims to determine the main topic structure with the words and weight values formed with the observed data set.

2.5 Logictic Regression

Logistic regression is a supervised classification algorithm (Liu et al., 2011). Instances in each category of the corpus are classified based on logistic function. Logistic regression is the appropriate regression analysis to be carried out when the dependent variable is binary. Logistic regression is an estimated analysis like it is in all regression analysis cases. Logistic regression is used to define the data and explain the relationship between a dependent binary variable and independent variables in a corpus.

3. Combining Labeled and Unlabeled Data with Semantic Values of Terms (CLUDS)

In this article, an original semi-supervised algorithm is presented, which uses two semantic smoothing kernels that both take advantages of semantic values of terms. CLUDS includes four sub-modules: preprocessing, instance labeling, combining labeled and unlabeled data, and prediction. These sub-modules are detailed in the following sub-sections.

Prior to using documents in both training and classification, some preprocessing and filtering are required to increase their suitability for computation and to remove content that is not useful for the computation. This preprocessing step is applied for all the data used in all test cases in this article.

Preprocessing for Long Textual Materials:
Firstly, stemming and stopword filtering are performed on the textual materials. Stemming is performed using Snowball(https://snowballstem.org/) stemmer, which is a small string-processing tool, designed for creating stems of the words. Snowball has implementations for both English and Turkish. Secondly, rare words, which are seen fewer than 3 times in documents, are filtered. Moreover, stopword filtering is also applied. Then, the most informative 2,000 words are selected using Information Gain (IG).

Preprocessing for Short Textual Materials:
For lemmatization, Zemberek-NLP (Akın & Akın, 2007) is used on Twitter dataset. Mentions, stickers, emoji, unnecessary words, numbers, and punctuation marks are removed from tweets. In order to create topical clusters on the data set, Mallet's (Graham et al., 2012) Latent Dirichlet Allocation (LDA) implementation is used.

All words in each Tweet in our Twitter corpus are randomly assigned topic labels by LDA algorithm. After topic assignment to tweets is completed, several statistics are extracted with this information. Local statistics show how many words are assigned to the topics in each tweet; while global statistics show how many times each word is

assigned to each topic for the entire tweet. After obtaining statistical information, each word is reassigned to each word for each tweet. The clustering analysis in topic-labels in Twitter dataset is shown in Figure 2.

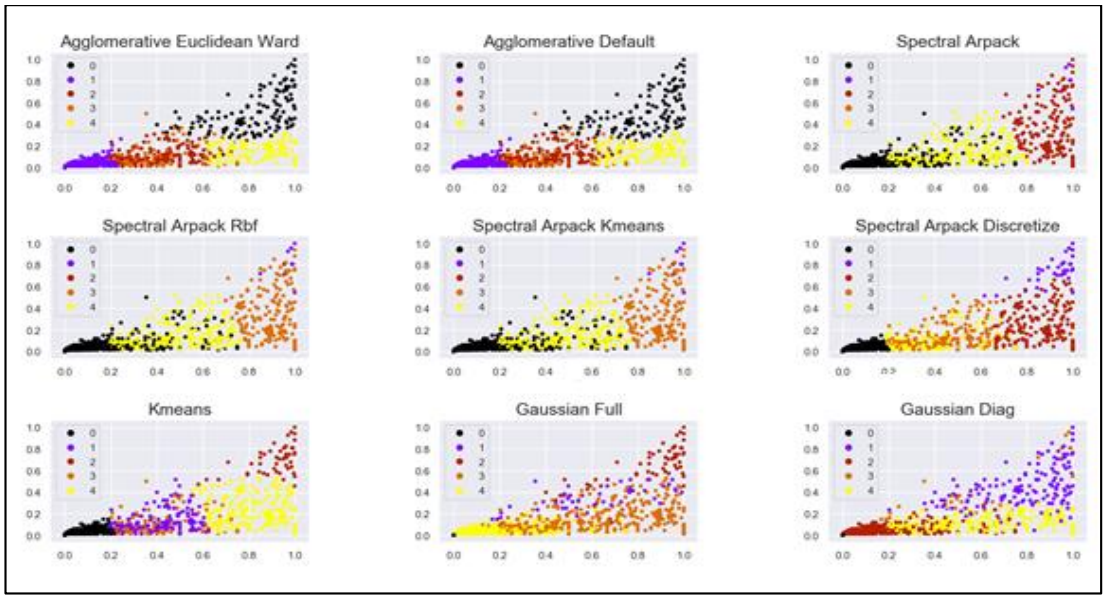

Figure 2. The clustering analysis in topic-labels in Twitter dataset

Instance Labeling:

In order to assign labels to unlabeled data in semi-supervised corpus CLUDS uses two methods: *1.) Relevance Values Kernel (RVK) and 2.) Logistic Regression.*

Relevance Values Kernel (RVK): Relevance Values Kernel (RVK) is used to label unlabeled documents and use them in the training phase of SVM. The aim of RVK is to benefit from the class-based term relevance weights in the semantic smoothing kernel building process by forming the semantic associations among words. RVK mainly depends on TF-RF, which is a supervised term weighting method as mentioned in Section 2.3. Inspired by TF-RF, it was decided to benefit from the class-based term relevance weights in the semantic smoothing kernel.

This relevance calculation contributes to the classifier since this type of weighting matrix has extra information related to the terms compared to BOW representation; these results expose semantic similarities among terms and instances by smoothing the demonstration of the textual materials.

To enrich the standard linear kernel function by semantic proximity between terms, the semantic proximity matrix *S* was generated using class-based term relevance weights as mentioned in (Lan et al., 2009). The class-based term relevance weighting calculations have been applied as in described in Section 2.3. Calculated weighting values of words were utilized in the kernel function as

$$S = RR^T, \tag{2}$$

where *R* is a class-based term relevance weights matrix, which is computed with Equation (1).

According to Equation (1) when a term is not seen in a class its weighting value for that class will be zero. After making all the computations, R is generated as a term-by-class matrix. *R* includes the relevance weights of terms for all classes.

In CLUDS, *S* is a semantic proximity matrix to convert documents from the input space to feature space. Mathematically, the semantically enhanced BOW demonstration of a document *d* is given as

$$\bar{\phi}(d) = \phi(d)\ S, \tag{3}$$

where *S* is the class-based semantic matrix, which represents the relevance values of words based on Equation (1) for each class by using the documents in the training set. The similarity value between two documents is calculated as:

$$k_{\mathrm{RVK}}(d_1, d_2) = d_1 S S^T d_2^T \tag{4}$$

Logistic regression is the appropriate regression analysis to be carried out when the dependent variable is binary. In our Tweet corpus, we have 5 classes. In order to make binary classification we conduct the experiments as one-to-many problem like taking one specific class and then treat the remaining classes as one big class.

Combining Labeled and Unlabeled Data:
In this step, the originally labeled documents (L_L) are combined with the documents labeled (L_P) by RVK at the previous *Instance Labeling* step and get *L*. An example scenario of this system is as follows: The number of labeled instances, unlabeled instances and test instances are 100, 7900 and 200; respectively. After assigning labels to these 7900 unlabeled instances by using RVK, the number of labeled instances is increased to 8000. This expanded set of documents, *L*, will be used as the training set for the previously published supervised classifier, CWK (Altınel et al., 2015).

Prediction:
In the prediction step, CLUDS attempts to predict the class labels of all the test samples. The architecture of CLUDS is shown in Figure 3. 20% of the dataset is separated as test portion and the remaining portion is divided as the labeled set and the unlabeled set. According to Figure 3, unlabeled documents are classified by RVK, and then these labeled documents are merged with originally labeled documents as it can be mentioned above. After that, CWK is built by this extended labeled set, and, finally, this model classifies test documents as it can be observed in Figure 3.

Sprinkled-CLUDS has also been implemented. Sprinkled-CLUDS has nearly the same architecture like CLUDS. The only difference between CLUDS and Sprinkled-CLUDS is that Sprinkled-CLUDS uses term-documents matrix augmented with sprinkled terms as mentioned in Section 2.4, while standard CLUDS uses original term-document matrix.

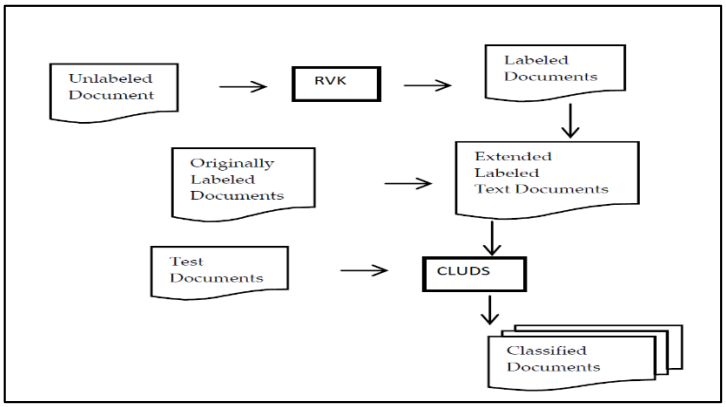

Figure 3. The Architecture of CLUDS

4. Experiment Setup

4.1 Data Collection:

In order to see the effect of CLUDS in different experiment setting, both normal text and short text materials are prepared and conducted into the test environment. For short textual material Twitter dataset is collected, for long textual material 3 different datasets (i.e., *20 Newsgroup dataset, 1150 Haber dataset, mini-newsgroups dataset*) are conducted into the experiment environment.

Twitter Dataset: This dataset is created at March of 2019 by collecting 1 M tweets under the following titles: *Science, Politics, Culture, Health, Sports.* Tweets are collected by using Twitter Streaming API. Additionally, Apache Spark, which uses Random Distributed Dataset (RDD) to process data at memory with many clusters, is used while collecting and processing the data. Since Spark runs in local memory, it is faster than Hadoop. Details of the dataset is given in Table 1.

20 Newsgroup Dataset: This dataset is a group of nearly 20,000 newsgroup documents under 20 classes. In this study, "COMP" subgroup of the 20 Newsgroup dataset is used. There are 5 classes in "COMP" dataset with 2500 instances and 2478 features. In this dataset, each class contains 500 documents (http:// www.cs.cmu.edu/~textlearning).

1150 Haber dataset: There are 1150 Turkish news-articles with 5 classes and 7948 features in this dataset. The categories are magazine, politics, sport, economy and health (Amasyalı & Beken, 2009).

mini-newsgroups dataset: It is also a subgroup of 20 Newsgroup datasets with 20 classes and there are 100 documents in each class. The number of total instances is 2000 with 12112 features (http://archive.ics.uci.edu/ml/).

Table 1. Details of Twitter Dataset

Topic Name	Number of tweets
Science	100000
Politics	300000
Culture	250000
Health	200000
Sports	150000

Twitter sentiment polarity dataset: In this study for sentiment detection, Turkish Tweet dataset is used. This dataset contains 29971 positive and 34233 negative tweets.

4.2 Experiment Setting and Evaluation

The testing set is kept as 20% of the corpus to evaluate the classification performance of the classifier. The class distributions in labeled, unlabeled and test percentages are similar to the class distributions in the original corpus. Classification accuracy is computed by taking the ratio of the number of correctly classified instances by the number of all test instances.

Sequential Minimal Optimization's (SMO) soft margin (C) parameter is set to 1(Kamber et al., 2005). All the algorithms are run 10 times on each training set level by arbitrarily choosing the documents to build the training corpus. Then the average of these 10 classification accuracies are calculated. Furthermore, in the results tables we also report standard deviations. Labeled data split in Figures 4-6 represent the percentage of labeled data, which is shown in Tables 2-5.

Additionally, for experiments on Twitter dataset, Scikit Learn's SVC class is used. Additionally Apache Spark ecosystem is used. Neo4J is used as data storage and calculations are performed by using its query language Cypher. All the experiments run on a machine with Intel i7-8750 @2.20Ghz CPU and 32GB RAM.

4.3 Baseline Algorithms

Linear kernel is the first baseline algorithm in this study. The second baseline algorithm is SSL-Linear. SSL-Linear first classifies unlabeled instances with linear kernel, which is trained by only the labeled samples. Then, it combines these labeled instances with the originally labeled instances and forms the classifier model by using customary linear kernel. After that, it again tries to classify unlabeled samples by the last generated model, compares the labels of each sample, and selects the estimations with higher classification confidence. The third baseline algorithm is previously published supervised semantic kernel, CWK (Altınel et al., 2015). The last baseline algorithm is RVK, which is detailed in Section 3.

5. Experiments, Evaluation Results and Discussion

5.1. Experiments with Long Texts

In the COMP dataset, CLUDS is better than all of the baseline kernels; linear kernel, SSL-Linear, CWK and RVK in all labeled set percentages as shown in Figure 4. For example, the classification performance gain of CLUDS over SSL-Linear kernel is about 21% at labeled set split 10% on COMP dataset according to Figure 4. This shows that the technique of CLUDS, which is based on combining labeled and unlabeled data by the semantics of terms, improves the classification performance on text classification field.

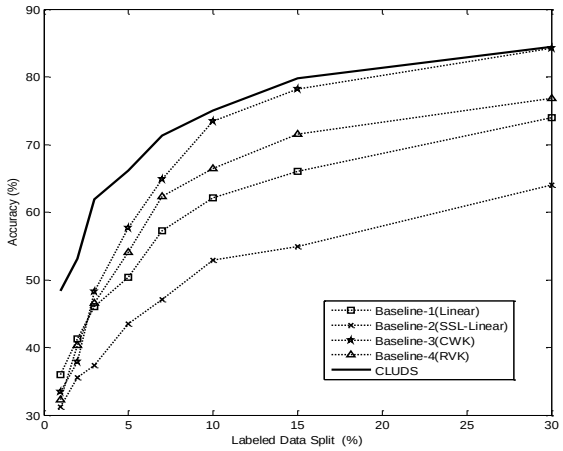

Figure 4. Classification Accuracies of Linear, SSL-Linear, CWK, RVK and CLUDS on COMP Dataset

Sprinkling process also seems to increase the classification accuracy since the classification accuracies of Sprinkled-CLUDS are higher than the classification accuracies of CLUDS at labeled set splits 1%, 2%, 3% and 10% according to Table 2.

Table 2. Classification Accuracies (%) of CLUDS and Sprinkled-CLUDS on COMP Dataset

Labeled %	Unlabeled %	CLUDS	Sprinkled-CLUDS
1	79	48.32±6.25	**48.73±1.43**
2	78	53.12±6.15	**54.12±2.11**
3	77	61.9±3.3	**62.01±2.54**
5	75	**66.14±5.02**	66.1±2.9
7	73	**71.28±2.23**	70.98±1.04
10	70	75.06±3.67	**75.11±0.05**
15	65	**79.82±2.21**	79.08±1.79
30	50	**84.46±1.55**	83.96±0.32

The same situation occurred for the 1150Haber dataset as shown in Figure 5. In 1150Haber dataset, the performance of CLUDS is superior to the performance of all other baseline algorithms at all labeled data splits. According to Figure 5, the classification performance of CLUDS is noticeably greater than all of the baseline algorithms at labeled set levels between 1% and 30% on 1150Haber dataset.

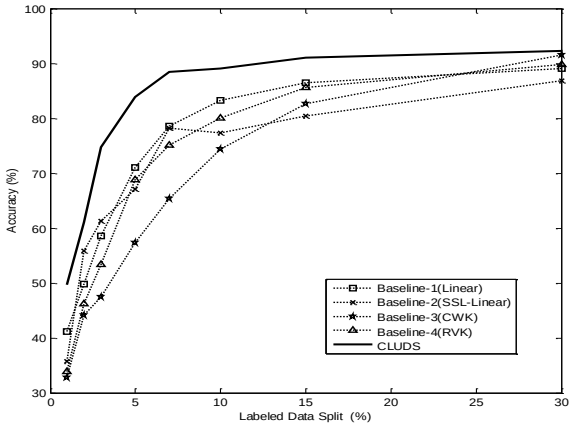

Figure 5. Classification Accuracies of Linear, SSL-Linear, CWK, RVK and CLUDS on 1150Haber Dataset

Sprinkling process also seems to advance the classification performance since the classification accuracies of Sprinkled-CLUDS are higher than the classification accuracies of CLUDS at most of the labeled set splits according to Table 3. This shows, the contribution of class labels into classification process, actually improves the classification capability of CLUDS.

Table 3. Classification Accuracies (%) of CLUDS and Sprinkled-CLUDS on 1150Haber Dataset

Labeled %	Unlabeled %	CLUDS	Sprinkled-CLUDS
1	79	49.7±9.76	**50.01±1.63**
2	78	**61.04±10.11**	60.94±2.23
3	77	74.7±4.12	**75.10±2.03**
5	75	83.91±5.09	**83.95±1.94**
7	73	88.43±2.95	**88.81±0.26**
10	70	89.09±1.63	**89.11±0.39**
15	65	**91±2.1**	90.79±0.29
30	50	**92.26±1.23**	92.10±2.45

The experimental results of all of the baseline algorithms and CLUDS in MiniNewsGroup dataset are shown in Figure 6. According to the results, CLUDS is superior to all other algorithms at every labeled set level in this dataset. For example, at labeled set level 2%, the classification accuracy of CLUDS is 55.4%, whereas the classification accuracies of linear, SSL-Linear, CWK and RVK are 38.42%, 31.68%, 32.67% and 8.93%, respectively.

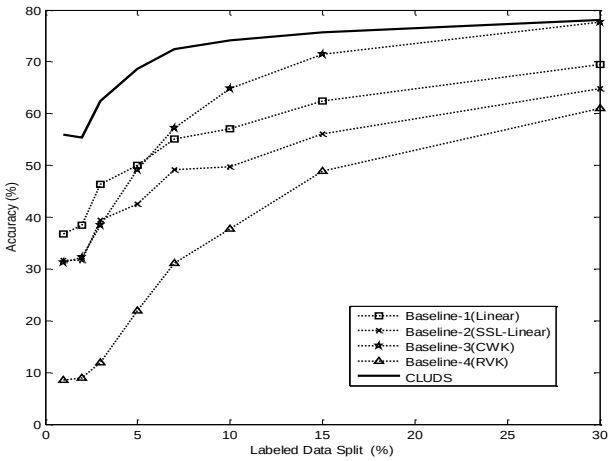

Figure 6. Classification Accuracies of Linear, SSL-Linear, CWK, RVK and CLUDS on MiniNewsGroup Dataset

5.2. Experiments with Short Texts

The experimental results of SSL-Linear, CLUDS, Sprinkled-CLUDS and Adaptive Sprinkled-CLUDS on Twitter dataset are shown in Table 4. According to the experimental results, the performance improvement of CLUDS over SSL-Linear is 8.53% that is a great performance progress since it is very hard to find labeled data in real world cases. Furthermore, Sprinkled-CLUDS and Adaptive Sprinkled-CLUDS algorithms are also run on Twitter dataset. Adaptive Sprinkling (Chakraborti et al., 2007) is a kind of Sprinkling where the number of class label contribution is decided based on the confusion matrix. For example if two classes are not separated with high classification accuracy by a classifier then the number of class label contribution will be high while if two classes are separated with high classification accuracy by a classifier then the number of class label contribution will be low. According to the experimental results, reported on Table 4, the classification accuracies of CLUDS, Sprinkled-CLUDS and Adaptive Sprinkled-CLUDS algorithms are 69.62%, 69.79% and 69.84% at labeled set 1% on Twitter dataset. This means that both Sprinkling and Adaptive-Sprinkling processes improve CLUDS algorithm from the point of classification performance. Additionally, according to Table 4; the classification performance gain of CLUDS over SSL-Linear is more noticeable especially at small labeled set splits, which is more valuable for real world problems since it is really very difficult to get labeled data.

Table 4. Classification Accuracies (%) of SSL-Linear, CLUDS, Sprinkled-CLUDS and Adaptive Sprinkled-CLUDS on Twitter Dataset

Labeled %	Unlabeled %	SSL-Linear	CLUDS	Sprinkled-CLUDS	Adaptive Sprinkled-CLUDS	Performance Improvement of CLUDS over SSL-Linear
1	79	61.09±1.12	69.62±1.98	69.79±2.13	69.84±3.18	**+8.53**
5	75	67.50±4.25	70.60±2.67	70.72±2.17	70.76±1.19	**+3.1**
10	70	69.62±2.56	70.78±1.45	71.63±1.89	71.67±2.35	**+1.16**
15	65	70.60±1.97	70.86±3.34	71.88±1.76	71.91±4.69	**+0.26**

5.3. Experiments with Twitter Sentiment Polarity Dataset

In order to see the effect of CLUDS on sentiment polarity detection task, Twitter sentiment polarity detection dataset is prepared with BOW format and classified with the classifiers. According to the experimental results, CLUDS is superior to SSL-Linear at every labeled set level in this dataset. For example, at labeled set level 1%, the performance improvement of CLUDS over SSL-Linear is 5.06% that is a great performance progress since it is very hard to find labeled data in real world cases.

Table 5. Classification Accuracies (%) of SSL-Linear and CLUDS on Twitter Sentiment Polarity Dataset

Labeled %	Unlabeled %	SSL-Linear	CLUDS	Performance Improvement of CLUDS over SSL-Linear
1	79	59.19±2.23	64.25±2.85	**+5.06**
5	75	63.52±3.21	68.60±2.75	**+5.08**
10	70	67.45±1.67	69.83±1.88	**+2.38**
15	65	69.63±2.89	71.64±2.14	**+2.01**

All the algorithms are run 10 times on each training set level by arbitrarily choosing the documents to build the training corpus. Then the average of these 10 classification accuracies are calculated. Furthermore, in the results tables we also report standard deviations. While the amount of labeled data increases in our experimental tables, we observe that the standard deviation behaves differently for different datasets and different test cases, sometimes increasing and sometimes decreasing. This is because the structure of each data set is different and there may be some noise generated when calculating the semantic values of the words by randomly selecting the training and test sets each time.

6. Conclusions and Future Work

Textual materials are broadly available on the Web. There are several kinds of methodologies that have been established for text classification. Nevertheless, text classification becomes more demanding when the textual materials have latent semantic connections. Extracting hidden semantic relationships from such unstructured form is a serious and hard task to perform. In this work, a novel semi-supervised semantic classification algorithm, CLUDS, is offered for text classification.

According to the experimental results, CLUDS effectively incorporates unlabeled samples into classifier. The highest gains of CLUDS over the remaining baseline algorithms are achieved in COMP dataset. For example, classification gains of CLUDS over SSL-linear are between 17.12% and 24.94% at labeled set levels 1%, 2%, 3%, 5%, 7%, 10%, 15%, 30%.

CLUDS is also applied on Twitter dataset since it seems superior to SSL-Linear at every labeled set level in this dataset. For example, at labeled set level 1%, the classification accuracy of CLUDS is 69.62%, whereas the classification accuracies of SSL-Linear is 61.09%. Consequently, the performance improvement of CLUDS over SSL-Linear is 8.53% that is a great performance progress since it is very hard to find labeled data in real world cases. To show the strength of the approach, it can be applied to further datasets.

Sprinkled-CLUDS and Adaptive-Sprinkled-CLUDS have also been implemented. Sprinkled-CLUDS has the same architecture like CLUDS with the only difference that is using extra sprinkling terms. This sprinkling process helps to extract latent class structures between documents. According to empirical evaluation results, sprinkling terms improve the classification accuracy in nearly most of the labeled set splits.

Acknowledgement

This work is supported in part by The Scientific and Technological Research Council of Turkey (TÜBİTAK) grant number 118E315 and grant number 120E187. Points of view in this document is hers of the author and do not necessarily represent the official position or policies of the TÜBİTAK.

Conflict of Interest

No conflict of interest was declared by the author.

References

Ahmed, I., Ali, R., Guan, D., Lee, Y., Lee, S., Chung, T. 2015. Semi-Supervised Learning Using Frequent Itemset and Ensemble Learning for SMS Classification. Expert Systems with Applications, 42(3), 1065-1073.

Akın, A. A., & Akın, M. D., 2007. Zemberek, an open source nlp framework for Turkish languages. Structure, 10, 1-5.

Alsmadi, I., & Hoon, G. K., 2019. Term weighting scheme for short-text classification: Twitter corpuses. Neural Computing and Applications, 31(8), 3819-3831.

Altınel, B., Diri, B., Ganiz, M.C., 2015. A Novel Semantic Smoothing Kernel for Text Classification with Class-based Weighting. Knowledge-Based Systems, 89(1), 265-277.

Altınel, B., Ganiz, M. C., 2018. Semantic Text Classification: A Survey of Past and Recent Advances. Information Processing & Management, 54(6), 1129-1153.

Amasyalı, M. F., Beken, A. Türkçe Kelimelerin Anlamsal Benzerliklerinin Ölçülmesi ve Metin Siniflandirmada Kullanilmasi, In Proceedings of IEEE Sinyal İşleme ve İletişim Uygulamalari Kurultayi (SIU), 2009.

Amor, B. R. , Vuik, S. I. , Callahan, R. , Darzi, A. , Yaliraki, S. N. , & Barahona, M., 2016. Community detection and role identification in directed networks: Understand- ing the twitter network of the care. data debate. In Dynamic networks and cyber.

Asiaee T, A., Tepper, M., Banerjee, A., & Sapiro, G., 2012. If you are happy and you know it... tweet. In Proceedings of the 21st ACM international conference on Information and knowledge management, 1602-1606.

Bai, X., Padman, R., Airoldi, E., 2004. Sentiment Extraction From Unstructured Text Using Tabu Search-Enhanced Markov Blanket. Carnegie Mellon University, School of Computer Science [Institute for Software Research International].

Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H. Greedy Layer-Wise Training of Deep Networks, 2007. Advances in Neural Information Processing Systems, 19(1), 153-160.

Biricik, G., Diri, B., Sönmez, A. C., 2009. A New Method for Attribute Extraction with Application on Text Classification, Soft Computing. Computing with Words and Perceptions in System Analysis, Decision and Control (ICSCCW), Fifth IEEE International Conference 2009, 1-4.

Biricik, G., Diri, B., Sönmez, A. C., 2012. Abstract Feature Extraction for Text Classification. Turkish Journal of Electrical Engineering & Computer Sciences, 2012, 20(1), 1137-1159.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. Journal of machine Learning research, 3(Jan), 993-1022.

Bloehdorn, S., Moschitti, A., 2007. Combined Syntactic and Semantic Kernels for Text Classification, Springer, 307-318.

Bordes, A., Glorot, X., Weston, J., Bengio, Y., 2012. Joint Learning of Words and Meaning Representations for Open-Text Semantic Parsing. In Proceedings of International Conference on Artificial Intelligence and Statistics, 127–135.

Blum, A. and Mitchell, T., 1998. Semi-Supervised Learning Literature Survey, In Proceedings of Conf. on Computational Learning Theory, 92-100.

Chakraborti, S., Lothian, R., Wiratunga, N., Watt, S. Sprinkling: Supervised Latent Semantic Indexing. In European Conference on Information Retrieval 2006, 510-514. Springer Berlin Heidelberg.

Chakraborti, S., Mukras, R., Lothian, R., Wiratunga, N., Watt, S. N., Harper, D. J. Supervised Latent Semantic Indexing Using Adaptive Sprinkling. In Proceedings of International Joint Conferences on Artificial Intelligence Organization (IJCAI), 2007, 7(1), 1582-1587.

Chapelle, O. and Zien, A., 2005. Semi-Supervised Classification by Low Density Separation, In Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, 57-64.

Chalothom, T., & Ellman, J., 2015. Simple approaches of sentiment analysis via ensemble learning. In information science and applications (pp. 631-639). Springer, Berlin, Heidelberg.

Chen, J., Huang, H., Tian, S., Qu, Y., 2009. Feature Selection for Text Classification with Naïve Bayes. Expert Systems with Applications, 36(3), 5432-5435.

Cho, Y. , Hwang, J. , & Lee, D., 2012. Identification of effective opinion leaders in the diffusion of technological innovation: A social network approach. Technological Forecasting and Social Change, 79 (1), 97–106.

Dahl, G., Ranzato, M., Mohamed, A-R., Hinton, GE., 2010. Phone Recognition with the Mean-Covariance Restricted Boltzmann Machine. In: Advances in Neural Information Processing Systems. Curran Associates, 469–477.

Dahl, G., Yu, D., Deng, L., Acero, A., 2012. Context-Dependent Pre-trained Deep Neural Networks for Large-Vocabulary Speech Recognition. IEEE Transactions of Audio Speech Language Processing, 20(1), 30–42.

Denecke, K., 2008. Using sentiwordnet for multilingual sentiment analysis. In 2008 IEEE 24th International Conference on Data Engineering Workshop, 507-512. IEEE.

Ferrara, E., Varol, O., Davis, C., Menczer, F., & Flammini, A., 2014. The rise of social bots. arXiv preprint arXiv: 1407.5225.

Fung, B.C.M., 2003. Hierarchical Document Clustering Using Frequent Itemsets, In Proceedings of International Conference on Data Mining, 59-70.

Graham, S., Weingart, S., & Milligan, I., 2012. Getting started with topic modeling and MALLET. The Editorial Board of the Programming Historian.

Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., and Kingsbury, B., 2012. Deep Neural Networks for Acoustic Modeling in Speech Recognition, IEEE Signal Processing Magazine, 29(6), 82-97.

Hinton, G., Osindero, S., Teh, Y-W., 2006. A Fast Learning Algorithm for Deep Belief Nets. Neural Computation, 18(7):1527–1554.

Hu, X., Tang, J., & Liu, H., 2014a. Online social spammer detection. In Twenty-Eighth AAAI Conference on Artificial Intelligence.

Hu, X., Tang, J., Gao, H., & Liu, H., 2014b. Social Spammer Detection with Sentiment Information. In 2014 IEEE International Conference on Data Mining (pp. 180-189). IEEE.

Hu, Y., Yi, Y., Yang, T., & Pan, Q., 2018. Short Text Classification with Convolutional Neural Networks Based Method. In 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV) (pp. 1432-1435). IEEE.

Injadat, M., Salo, F., & Nassif, A. B., 2016. Data mining techniques in social media: A survey. Neurocomputing, 214, 654-670.

Kalchbrenner, N., Grefenstette, E. and Blunsom, P., 2014. A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188.

Kamber, I.H., Frank, E. Data Mining: Practical Machine Learning Tools And Techniques, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

Kempe, D., Kleinberg, J., & Tardos, É., 2003. Maximizing the spread of influence through a social network. In Proceedings of the ninth acm sigkdd international conference on knowledge discovery and data mining (pp. 137–146). ACM.

Khan, F. H., Qamar, U., & Bashir, S., 2016. SentiMI: Introducing point-wise mutual information with SentiWordNet to improve sentiment polarity detection. Applied Soft Computing, 39, 140-153.

Koehler, M., Greenhalgh, S., & Zellner, A., 2015. Potential Applications of Sentiment Analysis in Educational Research and PracticeIs SITE the Friendliest Conference?. In Society for Information Technology & Teacher Education International Conference (pp. 1348-1354). Association for the Advancement of Computing in Education (AACE).

Krizhevsky A., Sutskever, I., Hinton, G., 2012. Imagenet Classification with Deep Convolutional Neural Networks.In: Advances in Neural Information Processing Systems. Curran Associates, 25(1), 1106–1114.

Lan, M., Tan, C. L., Su, J., Lu, Y. 2009. Supervised and Traditional Term Weighting Methods for Automatic Text Categorization. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(4), 721-735.

Liu YY, Yang M, Ramsay M, Li XS, Coid JW (2011) A comparison of logistic regression, classification and regression tree, and neural networks models in predicting violent re-offending. J Quant Criminol 27(4):547–553.

Luo, L., Yang, Y., Chen, Z., & Wei, Y., 2018. Identifying opinion leaders with improved weighted LeaderRank in online learning communities. International Journal of Performability Engineering, 14(2), 193-201.

Mikolov, T., Karafiat, M., Burget, L., Cernocky, J., and Khudanpur, S., 2011. Recurrent Neural Network Based Language Model, In Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 045–1048.

Mishne, G. and Glance, NS, 2006. Predicting movie sales from blogger sentiment," in AAAI 2006 Spring Symposium on Computational Approaches to Analyzing Weblogs.

Moore, A. Support Vector Machines, Tutorial slides, http://www.cs.cmu.edu/~awm, 2003.

Muslea, I., Minton, S., Knoblock, C.A., 2002. Active Semi-Supervised Learning In Robust Multi-View Learning. In Proceedings of the Nineteenth International Conference on Machine Learning.

Nakagawa, T. Inui, K. and Kurohashi, S., 2010. Dependency tree-based sentiment classification using CRFs with hidden variables. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 786–794. Association for Computational Linguistics.

Nigam, K., McCallum, A. K., Thrun, S., Mitchell, T., 2000. Text Classification From Labeled And Unlabeled Documents Using EM, Machine Learning, 39(2/3), 103-134.

Nigam, K., Ghani, R., 2000b. Analyzing the Effectiveness and Applicability of Co-Training. In Proceedings of the 9th ACM International Conference on Information and Knowledge Management, Washington, DC, 86–93.

Pang, B., Lee, L., & Vaithyanathan, S., 2002. Thumbs up?: sentiment classification using machine learning techniques. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10 (pp. 79-86). Association for Computational Linguistics.

Papka, R., Allan, J., 1998. Document Classification Using Multiword Features, In Proceedings of the Seventh International Conference on Information and Knowledge Management Table of Contents, Bethesda, Maryland, United States, 124–131.

Peng, F., Schuurmans, D., 2003. Combining Naive Bayes and n-Gram Language Models for Text Classification. In European Conference on Information Retrieval, 335-350. Springer Berlin Heidelberg.

Peng, Q., & Zhong, M., 2014. Detecting Spam Review through Sentiment Analysis. JSW, 9(8), 2065-2072.

Razon, A. R., Barnden, J. A., 2015. A New Approach to Automated Text Readability Classification based on Concept Indexing with Integrated Part-of-Speech n-Gram Features. Recent Advances in Natural Language Processing, 521-528.

Reborto, D. S., C., 2012 Kernel Functions for Machine Learning Applications, http://crsouza.com.

Rosenberg, C. et al., 2005. Semi-Supervised Self-Training of Object Detection Models, In Proc. 7th Workshop on Applications of Computer Vision, (1), 29-36.

Salah, Z., Al-Ghuwairi, A. R. F., Baarah, A., Aloqaily, A., Qadoumi, B. A., Alhayek, M., & Alhijawi, B., 2019. A systematic review on opinion mining and sentiment analysis in social media. International Journal of Business Information Systems, 31(4), 530-554.

Seide, F., Li, G., Yu, D., 2011. Conversational Speech Transcription Using Context-Dependent Deep Neural Networks. In Proceedings of International Symposium on Computer Architecture, 437–440.

Shinnou, H., Xiao, L., Sasaki, M., Komiya, K., 2015. Hybrid Method of Semi-supervised Learning and Feature Weighted Learning for Domain Adaptation of Document Classification, In Proceeding of the 29th Pacific Asia Conference on Language, Information and Computation, 496-503.

Silva, J., Coheur, L. Mendes, A.C. and Wichert, A., 2011. From symbolic to sub-symbolic information in question classification. Artificial Intelligence Review, 35(2):137–154.

Song, G., Ye, Y., Du, X., Huang, X., Bie, S., 2014. Short Text Classification: A survey, Journal of Multimedia, 9/5, 635-643.

Ucan, A., Naderalvojoud, B., Akcapinar Sezer, E. and Sever, H., 2016. SentiWordNet for New Language: Automatic Translation Approach. 12th International Conference on Signal-Image Technology & Internet-Based Systems.

Uysal, A. K., Gunal, S., 2014. Text Classification Using Genetic Algorithm Oriented Latent Semantic Features. Expert Systems with Applications, 41(13), 5938-5947.

Van Eck, P. S., Jager, W., & Leeflang, P. S., 2011. Opinion leaders' role in innovation diffusion: A simulation study. Journal of Product Innovation Management, 28(2), 187-203.

Wang, P., Xu, B., Xu, J., Tian, G., Liu, C. L., & Hao, H., 2016. Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification. Neurocomputing, 174, 806-814.

Wang, S. and Manning, C. ,2012. Baselines and bigrams: Simple, good sentiment and topic classification. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2, pages 90–94. Association for Computational Linguistics.

Yardi, S., Romero, D., & Schoenebeck, G., 2009. Detecting spam in a twitter network. First Monday, 15(1).

Yarowsky, D., 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In Proceedings of the 33rd

Annual Meeting of the Association for Computational Linguistics, 189–196.
Zeng, J., Li, J., Song, Y., Gao, C., Lyu, M. R., & King, I., 2018. Topic memory networks for short text classification. arXiv preprint arXiv:1809.03664.
Zhao, Y. , Li, S. , & Jin, F., 2016a. Identification of influential nodes in social net- works with community structure based on label propagation. Neurocomputing, 210, 34–44.
Zhao, Q. , Erdogdu, M. A. , He, H. Y. , Rajaraman, A. , & Leskovec, J., 2015. Seismic: A self-exciting point process model for predicting tweet popularity. In Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data min.
Zhou, X., Zhang, X., Hu, X., 2008. Semantic Smoothing for Bayesian Text Classification with Small Training Data. In Proceedings of International Conference on Data Mining, 289-300.
Zhu, X. J., 2005. Semi-supervised Learning Literature Survey, Technical Report, Department of Computer Sciences, University of Wisconsin at Madison, Madison, WI.