Journal of Universal Computer Science, vol. 24, no. 11 (2018), 1582-1603
submitted: 27/2/18, accepted: 7/9/18, appeared: 28/11/18 © *J.UCS*

Open Domain Targeted Sentiment Classification Using Semi-Supervised Dynamic Generation of Feature Attributes

Shadi Abudalfa
(King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia
shadi_abudalfa@hotmail.com, g201305390@kfupm.edu.sa)

Moataz Ahmed
(King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia
moataz@kfupm.edu.sa)

Abstract: Microblogging services have been significantly increased nowadays and enabled people to share conveniently their sentiments (opinions) with regard to matters of concerns. Such sentiments have shown an impact on many fields such as economics and politics. Different sentiment analysis approaches have been proposed in the literature to predict automatically sentiments shared in micro-blogs (e.g., tweets). A class of such approaches predicts opinion towards specific target (entity); this class is referred to as target-dependent sentiment classification. Another class, called open domain targeted sentiment classification, extracts targets from the micro-blog and predicts sentiment towards them. In this research work, we propose a new semi-supervised learning technique for developing open domain targeted sentiment classification by using fewer amounts of labelled data. To the best of our knowledge, our model represents the first semi-supervised technique that is proposed for open domain targeted sentiment classification. Additionally, we propose a new supervised learning model for improving accuracy of open domain targeted sentiment classification. Moreover, we show for the first time that SVM HMM is able to improve accuracy of open domain targeted sentiment classification. Experimental results show that our proposed technique outperforms other prominent techniques available in the literature.

Keywords: Text Mining, Social Opinions, Open domain, Targeted Sentiment Analysis, Polarity Classification, Semi-Supervised Learning.
Categories: H.3.3, I.2.1, I.2.2, I.2.4, I.2.6, I.2.7, I.7, L.3.2

1 Introduction

Sentiment Analysis [Liu, 2015] is an active research area nowadays and its importance is increased significantly with developing tasks of natural language processing (NLP). Sentiment analysis deals with mining opinions that are included in text for discovering insights [Chamlertwat, 2012]. Sentiment analysis has been employed in numerous applications whilst in this research we address specifically a task of identifying sentiment polarities in micro-blogs. The proposed approach covered in our work deals with detecting topics in the micro-blog and identifying sentiments toward them. This approach is referred to as open domain targeted sentiment classification.

The first approach of open targeted sentiment classification was proposed in 2013 by Mitchell et al. [Mitchell, 2013]. Their proposed approach joins two tasks: named entity recognition (NER) [Ratinov, 2009] and targeted sentiment classification. NER identifies named entities (targets) in the micro-blog. While, targeted sentiment classification predicts sentiment polarities toward identified targets. The accuracy of their proposed approach improved significantly with Spanish micro-blogs, whereas it is still limited with English micro-blogs. This limitation encourages us to fill this gap by improving accuracy of NER and sentiment prediction for English micro-blogs.

In this work, we employed methods of semantic web by using word embeddings [Mikolov, 2013] for improving accuracy of open domain targeted sentiment classification. Word embeddings is a method of substituting each token (word) in micro-blog by a numerical vector. Word embeddings preserve similarity between similar words in meaning. Thus, word embeddings convert words to vectors while the similarity between vectors mimics semantic similarity between words. As a result, word embeddings improved performance of many applications used in deep learning and NLP.

A recent survey by the authors [Abudalfa, 2017] has shown that there is no existing research that employs semi-supervised learning technique for open domain targeted sentiment classification. Thus, using supervised learning techniques for open domain targeted sentiment analysis needs a lot of labelled data during process training models. However, providing labelled micro-blogs is a difficult task since annotating micro-blogs is a time consuming process and usually leads to many errors that are related human mistakes. Even using automated systems for annotating micro-blogs is inaccurate. Additionally, using open domain targeted sentiment approach compounds the problem, since we need to provide labels for both included tasks: labels for NER and other ones for sentiment prediction. In this work, we address these issues by developing the first semi-supervised based technique for open domain targeted sentiment analysis.

The rest of this paper is organized as follows: Section 2 presents theoretical background for some related topics. Section 3 introduces a literature review for some related works. Section 4 illustrates all details of our proposed solutions. Section 5 describes the environment which is used for conducting our experiments. Section 6 shows all experimental results. Finally, Section 7 concludes the paper and presents suggestions for future work.

2 Background

Many techniques are proposed in the state of the art for detecting polarities expressed in micro-blogs. The formal approach is based on identifying sentiment (opinion) that is expressed toward the whole micro-blog. This approach cannot detect more than one sentiment even if the micro-blog talks about more than one topic (target). Thus, this approach is referred to as target-independent sentiment classification. Recently, some research works manipulate weakness of target-independent approach by predicting sentiment toward a specific target included in the micro-blog. This recent approach is referred to as target-dependent sentiment classification.

For example, when we try to analyze a micro-blog “Concorde is better than Boeing for long trips” by using target-independent approach, the predicated sentiment will be always “positive” sentiment since the micro-blog contains only positive phrase “better than”. While applying target-dependent sentiment classification will output “positive” sentiment if the interested target is “Concorde”, otherwise the output will be “negative” sentiment when the requested target is “Boeing”.

A more challengeable approach deals with predicting firstly the name entities (targets) in the micro-blog and then identifying sentiments toward them. Referring to the above example, the system will detect firstly words “Concorde” and “Boeing” as targets and then identify sentiments toward them as discussed previously. This very recent approach is referred to as open domain targeted sentiment analysis. Next subsection presents a theoretical background for implementing this approach. We conclude the section by describing some evaluation metrics used in this research direction.

2.1 Open Domain Targeted Sentiment

Since open domain targeted sentiment classification needs to identify all named entities in the micro-blog, it deals with NER for achieving this task. Then, we can determine which name entities represent the targets in the micro-blog. This process shifted the research direction from sentence level into word (token) level. Thus, we need to analyze a sequence of words (tokens) that forms each micro-blog. For classifying a sequence of words, we can use a main task that is used broadly in NLP called sequence labelling [Nguyen, 2007].

To sum up, open domain targeted sentiment classification represents each micro-blog as a sentence of tokens. Then, sequence labelling identifies all name entities that are related to persons, organizations, etc. One of the most famous strategies that are used by sequence labelling is called BIO. This strategy uses “B” tag to identify the beginning of named entity, or it labels “I” for determining tokens inside the named entity, otherwise the token will be labelled as “O” (outside) tag.

Sequence labelling can be developed by using hidden Markov model (HMM) [Altun, 2003] or conditional random field (CRF) [Keerthi, 2007]. In this research work, we use hidden Markov support vector machine (SVM HMM) for improving accuracy of open domain targeted sentiment classification. To the best of our knowledge, SVM HMM has not been used before in this research direction. SVM HMM is a model of sequence tagging with structural support vector machines by combining hidden Markov model with SVM. SVM HMM outperforms CRF based on many previous studies [Nguyen, 2007] [Keerthi, 2007]. Thus, it is interesting to employ it with open domain targeted sentiment classification.

2.2 Performance Evaluation

There are different metrics have been used for evaluating performance of open domain targeted sentiment classification in the state of the art. Since this research direction is a problem of sequence labelling, the classification accuracy is calculated by using two specific metrics. The first one is referred to as Acc-all which measures the accuracy of the entire named entity tags (including O labels) along with the sentiment tag. While the second specific metric is called Acc-Bsent which measures the accuracy of identifying the beginning of a named entity (B tags) together with the sentiment expressed towards it. We can use as well a metric called Zero/one-error for measuring percentage of micro-blogs that had at least one misclassified tag. Other traditional metrics can be used also in this research direction such as precision, recall, and F1-score [Parambath, 2014].

3 Literature Review

The approach of open domain targeted sentiment analysis is proposed firstly in 2013 by Mitchell et al. [Mitchell, 2013]. Their proposed approach is referred to as open domain targeted sentiment classification since it is capable to predict sentiments expressed in micro-bog for any named person or organization. Three models are proposed in this direction: pipeline, joint, and collapsed. Pipeline model identifies firstly named entities in the micro-blog then assigns sentiments toward them. Joint model identifies named entities along with their corresponding sentiments in one shot. In collapsed model, labels of named entity and sentiment polarity are combined in one label sequence.

These three models are compared against a baseline model where they use their volitional entity labels and assign no sentiment directed towards the entity (the majority case). They introduced strongly this baseline model to isolate how their methods perform specifically for the task of identifying sentiment targeted along with an entity. All proposed models are implemented by using CRF with a set of discrete features. During the same period, Klinger and Cimiano [Klinger, 2013] proposed also a close approach by employing factor graph for extracting both target entities and sentiment expressions.

A recent work [Zhang, 2015] employed neural networks instead of CRF for improving accuracy of open domain targeted sentiment classification. Efficiency of using word embeddings (neural features) is evaluated in comparison with using discrete features that are used by Mitchell et al. This work is evaluated by using same dataset collected by Mitchell et al. for making comparisons. The reported results show that using both neural and discrete together improved significantly performance of open domain targeted sentiment classification.

A very recent work [Li, 2017] proposed a new model for improving performance of open domain sentiment classification. The proposed solution is based on building a graphical model for extracting both named entities and their associated sentiment polarities by using collapsed strategy. This model is validated by using same dataset collected by Mitchell et al. to make comparisons with both previous works [Mitchell, 2013] [Zhang, 2015]. The feature engineering is based on using same discrete features that are used by Mitchell et al. and Zhang et al. in addition to obtaining more polarity information by using same lexicons used by Mitchell et al. Reported results show that this graphical model outperforms all models that are proposed by previous related works [Mitchell, 2013] [Zhang, 2015].

There are some related works that are close in spirit to open domain targeted sentiment analysis such as researches achieved by Hu et al. [Hu, 2004] and Popescu et al. [Popescu, 2007]. These research works are based on aspect-oriented sentiment analysis which extracts product attributes from user reviews and predict opinions towards them. Moreover a topic-oriented sentiment analysis is proposed by Wang et al. [Wang, 2011] for extracting features and sentiments towards certain topics.

4 Proposed Solutions

In this research direction, we propose three new solutions for improving accuracy of open domain targeted sentiment classification. The first proposed solution is based on combining discrete features with multiple word embeddings. The second solution is based on employing semi-supervised learning by generating feature attributes dynamically. The last proposed solution combines supervised learning with dynamic generation of feature attributes. In the subsections to follow, we describe these solutions in details.

4.1 Supervised Learning of Combined Discrete Features and Multiple Word Embeddings

Using word embeddings increases accuracy of open domain targeted sentiment classification significantly [Zhang, 2015]. The problem of using word embeddings with social media lies in finding numeric vector that represents each word in the micro-blog. Logically, it is impossible to provide word embeddings for representing each word in micro-blog since bloggers usually use slang words when writing their micro-bogs. In the best case, we can provide word embeddings for representing each word in micro-blog during training the machine learning model. But we cannot find word embeddings for representing each word in micro-blogs when testing the model since we cannot know all words that are used by bloggers in real life situation. Of course, existence of missed word embeddings limits accuracy of any machine leaning model.

To decrease effect of these missed words, we propose a solution that is based on compiling pre-trained word embeddings form different resources. In this solution, we concatenate pre-trained word embedding from different sources. Thus, probability of missing word embeddings when representing strange words will be decreased. To increase accuracy of our proposed model, we normalized all concatenated word embeddings to be fallen in the same range. Additionally, we concatenated all word embeddings with discrete feature attributes for increasing performance of this solution.

Our proposed solution uses SVM HMM model to take into consideration the relations between words of each sequence in micro-blogs. We selected this machine learning model because this research direction is represented as a sequence labelling problem and it is improper to use traditional SVM. Based on our knowledge, our research is the first work that employs SVM HMM for improving performance of open domain targeted sentiment classification. One more advantage of using SVM HMM is that it accepts numerical (continuous) or categorical (discrete) features or a combination of them. All details of training our proposed model are illustrated in Figure 1. We normalized each vector that represents corresponding micro-blog by applying next formula which normalizes all numeric values to be fallen in the range between -1 and 1.

$$X_{new} = 2\left(\frac{X - X_{\min}}{X_{\max} - X_{\min}} - \frac{1}{2}\right) \tag{1}$$

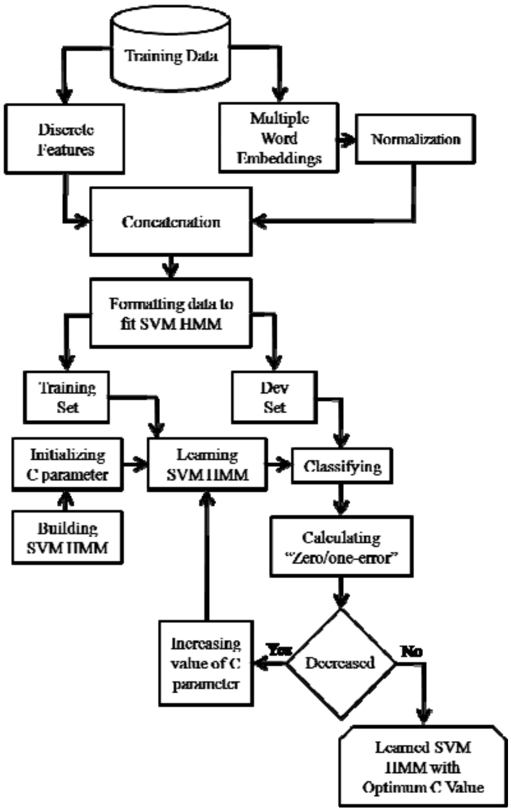

Figure1: Flowchart of training model for combining discrete features with multiple word embeddings

We use optimization method to find optimum value of *C* parameter. The optimization process is conducted by increasing value of *C* parameter gradually. At each selected value of *C* parameter, we test the model with the development set and calculate "zero/one-error" metric. We selected the optimum value of *C* parameter that is provided by the lowest value of "zero/one-error" metric. Of course, using testing set instead of using development set will provide better optimum value of *C* parameter. But, we use development set in this optimization process to make our proposed solution more realistic. In real problem, we cannot see testing data while we can use development data (which is a part of training data) for testing.

When classifying the new unseen micro-blog, we use the trained SVM HMM model which is learned by using the optimum value of *C* parameter. To check efficacy of our proposed solution we can apply it on the testing data and calculate evaluation metrics for name entity recognition (NER) and sentiment analysis (SA). The most common metrics that are used for evaluating open domain targeted sentiment classification are precision, recall, and F1-score.

All details of testing our proposed model are illustrated in Figure 2. Of course, we need also to collect word2vec embeddings of each word in testing data from the same sources that are used with training model. Then we need to concatenate the multiple word2vec embeddings with the discrete features as illustrated in the figure. Finally, we need to convert data form to fit format used with SVM HMM model (as used with training model) to be ready for classification.

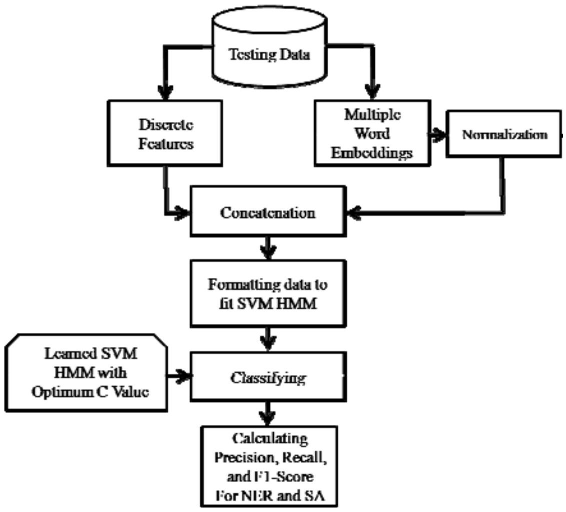

Figure 2: Flowchart of testing model for combining discrete features with multiple word embeddings

4.2 Semi-Supervised Learning with Dynamic Generation of Feature Attributes

We propose here a new technique for employing semi-supervised learning in open domain targeted sentiment classification by using both labelled and unlabeled data [Chapelle, 2006]. Based on our knowledge, our solution is the first semi-supervised learning technique that is proposed for open domain targeted sentiment classification. Our proposed solution is based on improving accuracy by generating more attributes to the horizontal level of each word (token). Thus, our solution adds more attributes for each feature vector that represents each word (token) in each micro-blog. Our proposed solution works on level of feature attributes since evaluating open domain targeted sentiment analysis is based on word level instead of micro-blog level.

Using traditional semi-supervised learning techniques is not suitable for open domain targeted sentiment classification because these techniques ignore the relations between words of each sequence in micro-blogs. Thus, our semi-supervised based solution is more suitable for this research direction in comparison with other approaches that deal with micro-blog level such as self-learning and semi-supervised text classification by using expectation maximization [Nigam, 2006]. Our proposed solution is inspired by approach proposed by Qi et al. [Qi, 2009]. However, we developed in this work a new method for generating feature attributes. Our proposed method is simpler and decreases time consumed for generating feature attributes. Figure 3 describes the algorithm of our new semi-supervised learning method.

Algorithm of new semi-supervised learning technique for open domain targeted sentiment classification:

Inputs: Label *ratio*, training set (*trainSet*), Development set (*DevSet*), testing set (*TestSet*)
Output: precision, recall, and F1-score of classifying testing data

1) Split *trainSet* into labelled data (*trainSetLab*) equals *ratio* value and the rest as unlabeled data (*trainSetUnLab*)
2) Build SVM HMM model and train it by using *trainSetLab* data with an initial small value of C parameter
3) Calculate zero/one-error of classifying *DevSet*
4) Increase value of C parameter and repeat steps 2 and 3 until zero/one-error does not decrease.
5) Check performance of SVM HMM model by using optimum value of C parameter.
6) Select only numeric values in each vector of *trainSetUnLab* data and store them in *trainUnLabArray*
7) Cluster the *trainUnLabArray* by using k-means with initial value of number of clusters (*ClusterNum*).
8) For each word in *trainSetLab* determine cluster ID (*ClusterID*) which the word belongs to.
9) Normalize values of all *ClusterID* to form *ClusterIDNorm* for each word in *trainSetLab* data .
10) Concatenate *ClusterIDNorm* as new feature attribute to the feature vector of each word in *trainSetLab* to form *trainSetLab+*.
11) Retrain the SVM HMM model by using *trainSetLab+*.
12) Increase value of *ClusterNum* and iterate steps 5 to 10 until stopping criterion is met.
13) Classify *TestSet* data by using the best SVM HMM model and output results.

Figure 3: A new semi-supervised learning technique for open domain targeted sentiment classification

The optimization process, which is conducted by using steps 2 and 3, is the same optimization method which is illustrated in Figure 1 for finding optimum value of *C* parameter. In step 6, we select only numeric values which represent neural features and skip discrete attributes for improving accuracy of data clustering. In step 7, we use k-means [Abudalfa, 2013] algorithm for clustering unlabelled data while number of clusters is increased iteratively in step 12. The normalization process in step 9 is calculated by dividing each cluster id (ClusterID) by total number of clusters (ClusterNum). Thus, the values of normalized cluster ids (ClusterIDNorm) are fallen in the range (0, 1]. This normalization process makes values of the new generated attributes close to values of other neural features included in the dataset. As a result, the samples will be more discriminated and classification accuracy will be improved. The stopping criterion in step 12 can be conducted by using different ways. In this work, we achieved stopping criterion by checking whether the performance of learned SVM HMM (step 5) does not improve after increasing value of ClusterNum.

4.3 Supervised Learning with Dynamic Generation of Feature Attributes

This solution is similar to solution of "semi-supervised learning with dynamic generation of feature attributes" which is represented in Figure 3. But we use all training set (trainSet) as labelled data instead of splitting it into labelled and unlabelled data when training the SVM HMM model. We propose this solution to

check efficacy of applying method of dynamic generation of feature attributes with supervised learning technique.
To save memory and make this technique fast, we selected half of training data for conducting clustering process when generating feature attributes. The output of this technique is calculated by selecting the maximum achieved performance when applying incremental generation of feature attributes. If the generated feature attributes does not improve performance, then we take accuracy of burly supervised learning model.

5 Experiment Setup

We developed many experiments to test efficiency of using semi-supervised for target-dependent sentiment analysis. All experiments were carried using hardware and software tools which are implemented in information & computer science department at KFUPM. The measurement tools and hardware platform specifications are described in Tables 1, and 2 respectively.

Tool	Ver	Purpose
Python	2.7	Extracting Features, building and learning models for developing experiments, classifying micro-blogs, and computing results
Anaconda	4.2.0	Open data science platform powered by Python for providing development environment that facilitates developing our experiments
Spyder	2.3.8	Graphical platform for editing, testing and debugging Python codes
MS Excel	2016	Analyzing data
Vim	7.4	Text editor for editing huge training and testing data files

Table 1: Tools and programs

Component	Specification
CPU	Intel(R) Core (TM) i7-3720 3.40 GHZ
Memory	8.00 GB
OS	Windows 8 (64-bit)

Table 2: Platform specifications

Our experiments are conducted by using corpus (dataset) that is collected originally by Mitchell et al. [Mitchell, 2013] which is available publicly[1]. This corpus is used also by other related works [Zhang, 2015] [Li, 2017] that are compared in our research work. Thus, using this corpus enables us to make real comparisons with previous related works. The corpus includes both English and Spanish tweets where each word (token) is located in a separated line. Table 3 shows statistics of the corpus as illustrated in research work achieved by Zhang et al. [Zhang, 2015]. The corpus

[1] http://www.m-mitchell.com/code/index.html

consists of 10 folds and each fold is divided into training, testing, and development (dev) sets.

Domain	#Sent	#Entities	#+	#-	#0
English	2,350	3,288	707	275	2,306
Spanish	5,145	6,658	1,555	1,007	4,096

Table 3: Dataset for open domain targeted sentiment classification

5.1 Formatting Data and Feature Engineering

To be able to use the same public dataset utilized by pervious research works, we reformatted the feature vectors to fit our proposed models. We converted the data form which is included in implementation code developed by Zhang et al. [Zhang, 2015] to fit format used by SVM HMM[2]. We prepared the data to represent collapsed labels (b-negative, b-neutral, b-positive, i-negative, i-neutral, i-positive, and o).
As a result of this work, we prepared numerous datasets as described briefly in Table 4. We include only discrete features for checking performance of using these features alone. We refer to this resulted dataset as “Discrete_Data”. We used same discrete features that are generated by Mitchell et al. [Mitchell, 2013] and used by Zhang et al. [Zhang, 2015] and Li et al.[Li, 2017] as shown in Table 5.

We also prepared data that includes only features attributes of pre-trained word2vec embeddings provided by Zhang et al. [Zhang, 2015]. We refer to this data as “Word2VecZhang” which include feature vector of size 100 attributes. Then we normalized the “Word2VecZhang” and called it “Word2VecZhangNorm”. We prepared as well a dataset that combines both discrete and normalized word2vec embeddings to check its efficiency in increasing performance. We refer to this merged dataset as “Discrete_Word2VecZhangNorm”.

Additionally, we prepared data that includes pre-trained wor2vec embeddings provided by Al-Rfou et al. [Al-Rfou, 2013] which are used by Li et al. [Li, 2017]. These wor2vec embeddings are available online and can be downloaded freely[3]. Each vector of this word2vec embeddings contains 64 numeric values. The resulted dataset is called “Word2VecPolyglot” and its normalized version is called “Word2VecPolyglotNorm”. We merged as well these normalized word2vec embeddings with the discrete features and called it “Discrete_Word2VecPolyglotNorm”. We merged “Word2VecPolyglotNorm” and “Word2VecZhangNorm” to build data that includes both representations of word2vec embeddings. The combined version is called “Word2VecBothPolyglot&ZhangNorm” and the dataset which includes additionally discrete features is called “DiscW2VPolyglot&ZhangNorm”.

[2] https://www.cs.cornell.edu/people/tj/svm_light/svm_hmm.html

[3] https://sites.google.com/site/rmyeid/projects/polyglot

Dataset	Description
Discrete_Data	Includes only discrete features that are used by Mitchell et al. [Mitchell, 2013]
Word2VecZhang	Includes only word2vec embeddings features that are included by Zhang et al. [Zhang, 2015]
Word2VecZhangNorm	Normalized version of "Word2VecZhang" dataset
Discrete_Word2VecZhangNorm	Combines both "Discrete_Data" and "Word2VecZhangNorm" dataset
Word2VecPolyglot	Includes wor2vec embeddings which are used by Li et al. [Li, 2017]
Word2VecPolyglotNorm	Normalized version of "Word2VecPolyglot" dataset
Discrete_Word2VecPolyglotNorm	Combines both "Discrete_Data and Word2VecPolyglotNorm"
Word2VecBothPolyglot&ZhangNorm	Combines both "Word2VecPolyglotNorm" and "Word2VecZhangNorm" datasets
DiscW2VPolyglot&ZhangNorm	Combines both "Discrete_Data" and "Word2VecBothPolyglot&ZhangNorm" datasets
Word2VecBojanowski	Includes wor2vec embeddings of the third source [Bojanowski, 2017]
Word2VecBojanowskiNorm	Normalized version of "Word2VecBojanowski" dataset
Discrete_Word2VecBojanowskiNorm	Combines both "Discrete_Data" and "Word2VecBojanowskiNorm"
W2VpolyglotZhangBojanowskiNorm	Combines "Word2VecZhangNorm", "Word2VecPolyglotNorm", and "Word2VecBojanowskiNorm"
DW2VpolyglotZhangBojanowskiNor	Combines both "Discrete_Data" and "W2VPolyglotZhangBojanowskiNorm"

Table 4: Summary of all prepared datasets

Moreover, we prepared another form of data that includes a third source of pre-trained word embeddings called fastText [Bojanowski, 2017]. This representation of word2vec embeddings has dimension equals 300 attributes and it is available online[4]. The resulted dataset is called "Word2VecBojanowski" and the normalized version is called "Word2VecBojanowskiNorm". We merged also these normalized word2vec embeddings with the discrete features and called it as "Discrete_Word2VecBojanowskiNorm". We merged all three sources of word2vec embeddings in one dataset called "W2VPolyglotZhangBojanowskiNorm". When combining the discrete features to "W2VPolyglotZhangBojanowskiNorm", the resulted dataset is called "DW2VPolyglotZhangBojanowskiNor".

6 Results and Analysis

We developed many experiments for proving efficacy of proposed solutions. Next subsections describe our experiments and provide summary of our results. Discussions and analysis of experimental results are included as well in this section.

[4] https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md

Surface Features
binned word length, message length, sentence position; Jerboa features; word identity; word lengthening; punctuation characters, has digit; has dash; is lower case; is 3 or 4 letters; first letter capitalized; more than one letter capitalized, etc.
Linguistic Features
function words; can syllabify; curse words; laugh words; words for good/bad; slang words; abbreviations; intensiers; subjective suffixes and prefixes (such as diminutive forms); common verb endings; common noun endings
Brown Clustering Features
cluster at length 3; cluster at length 5
Sentiment Lexicon Features
is sentiment-bearing word; prior sentiment polarity

Table 5: Discrete features used in our work.

6.1 Using Cluster IDs as Feature

In this section, we describe our work for improving performance of open domain target sentiment models that are proposed by Mitchell et al. [Mitchell, 2013]. These models have been introduced as the first approach for open domain sentiment classification. Since we could not use numerical features with CRF model, we clustered the data to different clusters and used cluster ids (integer values) as additional feature attribute. We added this feature to the discrete feature attributes that are used by Mitchell et al. [Mitchell, 2013]. The added feature attribute represents the cluster which covers the corresponding word in the corpus.

To achieve our goal, we firstly found word embeddings that are representing each word in the used corpus by using pre-trained word2vec embeddings provided by Zhang et al. [Zhang, 2015]. Then, we clustered the data of word2vec embeddings for all entities in each tweet in the used corpus. Finally, we used cluster ids as additional feature attribute to the other discrete feature attributes that are used by Mitchell et al. [Mitchell, 2013].

We applied this method to the 2nd fold of corpus which is available in implementation code developed by Mitchell et al. [Mitchell, 2013]. We used k-means clustering algorithm for clustering all word2vec embeddings. Number of these word2vec embeddings (includes both training and testing data) is 35681 vectors. After adding cluster IDs as feature attribute to the used dataset, we checked efficiency of adding these attributes by training and testing all models proposed by Mitchell et al. [Mitchell, 2013]. We conducted this experiment by modifying implementation code developed by Mitchell et al. [Mitchell, 2013] which is available publically[5]. Table 6 describes all tested models while Table 7 shows results of our experiments when using cluster granularity that is equal to 0.1%. In this experiment, we used acc-all and acc-Bsent metrics to compare our work with similar work provided by Mitchell et al. since they used these metrics.

Based on the reported results, we can note that using cluster ids as additional feature attributes increases significantly performance of open domain targeted

[5] http://www.m-mitchell.com/code/index.html

sentiment classification. We can notice clearly that Collapsed_Clusters_Base model outperforms all other models with respect to Acc-all metric. While, Pipeline_Clusters_Base model outperforms all other models with respect to Acc-Bsent metric. This means that collapsed models are the best in general. But when our interested focuses a well on accuracy of name entity recognition, then our choice should be pipeline models. We can notice also that results of Acc-Bsent is too low (does not exceed 40%) since it is difficult to classify correctly the beginning of targeted entities.

In this proposed solution, we use sequence tagging with structural support vector which is referred to as SVM HMM[6]. We selected this machine learning technique because this research direction solves as sequence labelling problem and it is improper to use traditional SVM. Additionally, SVM HMM model accepts numerical (continuous) or categorical (discrete) features or a combination of them.

Model	**Description**
Joint_CRF_Base	Baseline joint model which uses volitional entity labels that are specified by Mitchell et al. [Mitchell, 2013] and assign no sentiment directed towards the entity.
Joint_CRF	Joint model proposed by Mitchell et al. [Mitchell, 2013]
Joint_Clusters_Base	Adding clusters ids as feature attribute to Joint_CRF_Base model
Joint_Clusters	Adding clusters ids as feature attribute to Joint_CRF model.
Pipeline_CRF_Base	Baseline pipeline model which uses volitional entity labels that are specified by Mitchell et al. [Mitchell, 2013] and assign no sentiment directed towards the entity.
Pipeline_CRF	Pipeline model proposed by Mitchell et al. [Mitchell, 2013]
Pipeline_Clusters_Base	Adding clusters ids as feature attribute to Pipeline_CRF_Base
Pipeline_Clusters	Adding clusters ids as feature attribute to Pipeline_CRF
Collapsed_CRF_Base	Baseline collapsed model which uses volitional entity labels that are specified by Mitchell et al. [Mitchell, 2013] and assign no sentiment directed towards the entity.
Collapsed_CRF	Collapsed model proposed by Mitchell et al. [Mitchell, 2013]
Collapsed_Clusters_Base	Adding clusters ids as feature attribute to Collapsed_CRF_Base
Collapsed_Clusters	Adding clusters ids as feature attribute to Collapsed_CRF

Table 6: Description of all evaluated models

To make our comparison with previous related works more accurate and fair enough, we used the same code that is provided by Li et al. [Li, 2017] for calculating evaluation metrics. Firstly, we need to apply an optimization task for selecting best value of *C* parameter when using SVM HMM model. It is important to clarify that we did not optimize epsilon parameter since using its default value is enough to converge to optimum accuracy while changing *C* parameter. We select the best value of *C* parameter that provides lowest "zero/one-error" when classifying dev set. The

[6] https://www.cs.cornell.edu/people/tj/svm_light/svm_hmm.html

evaluation metric "zero/one-error" is one of results that are provided by the used tool when building SVM HMM model. "zero/one-error" metric calculates the percentage of sentences (tweets) that had at least one misclassified tag (label).

Model		**NER/SA**	
		Acc-all	**Acc-Bsent**
Joint	Joint_CRF_Base	87.25	32.69
	Joint_CRF	87.18	32.05
	Joint_Clusters_Base	**90.18**	**33.83**
	Joint_Clusters	89.89	31.84
Pipeline	Pipeline_CRF_Base	87.73	32.01
	Pipeline_CRF	87.73	32.01
	Pipeline_Clusters_Base	**90.3**	**37.38**
	Pipeline_Clusters	90.06	35
Collapsed	Collapsed_CRF_Base	89.77	30
	Collapsed_CRF	89.77	30
	Collapsed_Clusters_Base	**90.44**	**32.41**
	Collapsed_Clusters	90.44	31.66

Table 7: Results of evaluating models

We trained the SVM HMM model by using different values of *C* parameter in the range between 1 into 550 with an increasing step that is equal to 10. With each selected *C* value we trained the SVM HMM model by using training data and calculated "zero/one-error" by classifying dev data. Finally, we use the best *C* value for classifying the testing data and calculating evaluation metrics (Precision, Recall, and F1-Score). It worth to clarify that using testing data instead of dev data will provide more optimum value of *C* parameter. But we use dev data rather than testing data to make our results more realistic.

We applied SVM HMM to the 2^{nd} fold of all prepared data for English tweets. We reported all results when using each dataset described in previous section as shown in Table 8. The maximum values in this table are highlighted as bold font. Experimental results show that there are 324 samples which match criteria of open domain targeted sentiment. These samples identify number of words (tokens) that are targeted as topics and have sentiments.

Since DW2VPolyglotZhangBojanowskiNor dataset provides the best results (lowest error) as shown in Table 8, we applied SVM HMM model to all folds of this dataset. All results provided by using both English and Spanish are reported in Table 9. This experiment uses optimization method to find best value of *C* parameter that is provided by the lowest value of "zero/one-error" (Err). We changed value of *C* parameter from 1 into 550 with increase step equals 10. The table includes also number of observed samples (obs) and number of samples (Pred) that are predicted correctly. The results include evaluations metrics of precision (P), recall (R), and F1-score (F1) for both name entity recognition (NER) and sentiment analysis (SA). The maximum values of classification accuracy and F1-score among all folds are highlighted by using bold and underlined font.

Dataset	Err	C	NER			SA		
			P	R	F1	P	R	F1
Discrete_Data	80.66	111	69.57	34.57	46.19	55.9	27.78	37.11
Word2VecZhang	91.51	101	57.58	17.59	26.95	43.43	13.27	20.33
Word2VecZhangNorm	91.98	101	50.85	18.52	27.15	37.29	13.58	19.91
Discrete_Word2VecZhangNorm	75.47	81	64.5	45.99	53.69	48.48	34.57	40.36
Word2VecPolyglot	82.55	41	67.88	34.57	45.81	51.52	26.23	34.76
Word2VecPolyglotNorm	82.55	41	65.73	36.11	46.61	50.56	27.78	35.86
Discrete_Word2VecPolyglotNorm	75.47	131	72.22	56.17	63.19	55.95	43.52	48.96
Word2VecBothPolyglot&ZhangNorm	79.25	41	66.15	39.2	49.22	51.56	30.56	38.37
DiscW2VPolyglot&ZhangNorm	73.11	31	71.68	50	58.91	54.87	38.27	45.09
Word2VecBojanowski	75	71	65.91	44.75	53.31	49.09	33.33	39.71
Word2VecBojanowskiNorm	75	81	68.64	46.6	55.51	51.82	35.19	41.91
Discrete_Word2VecBojanowskiNorm	74.06	41	73.84	54.01	62.39	54.85	40.12	46.35
W2VPolyglotZhangBojanowskiNorm	73.58	31	69.55	47.22	56.25	51.36	34.88	41.54
DW2VPolyglotZhangBojanowskiNor	**70.75**	**21**	**74.38**	**55.56**	**63.6**	**56.61**	**42.28**	**48.41**

Table 8: Summary of best result when applying SVM HMM to the 2nd fold of prepared datasets

Lang	Fold	Err	C	Obs #	Pred #	NER			SA		
						P	R	F1	P	R	F1
Eng	1	69.34	101	347	311	69.45	62.25	**65.65**	49.52	44.38	46.81
	2	70.75	21	324	242	74.38	55.56	63.6	56.61	42.28	48.41
	3	68.87	51	346	274	67.15	53.18	59.35	48.18	38.15	42.58
	4	73.11	51	318	253	67.59	53.77	59.89	49.41	39.31	43.78
	5	69.34	61	340	259	67.18	51.18	58.1	48.65	37.06	42.07
	6	68.87	31	319	243	72.43	55.17	62.63	51.85	39.5	44.84
	7	67.92	31	309	218	70.64	49.84	58.44	50.0	35.28	41.37
	8	69.34	21	320	233	74.68	54.37	62.93	60.09	43.75	**50.63**
	9	69.34	61	346	295	69.15	58.96	63.65	45.76	39.02	42.12
	10	69.81	31	319	232	68.1	49.53	57.35	48.71	35.42	41.02
	Avg	**69.67**	**46**	**329**	**256**	**70.08**	**54.38**	**61.16**	**50.88**	**39.42**	**44.36**
Span	1	64.87	81	677	556	77.16	63.37	69.59	50.54	41.51	45.58
	2	64.36	121	656	563	74.96	64.33	69.24	46.36	39.79	42.82
	3	62.42	151	676	524	75.38	58.43	65.83	50.19	38.91	43.83
	4	65.52	121	641	538	79.0	66.3	72.09	52.23	43.84	47.67
	5	64.58	111	669	545	81.28	66.22	**72.98**	51.56	42.0	46.29
	6	64.66	121	663	556	74.1	62.14	67.6	48.38	40.57	44.13
	7	65.44	141	651	533	76.17	62.37	68.58	47.28	38.71	42.57
	8	65.3	111	681	592	73.82	64.17	68.66	46.62	40.53	43.36
	9	62.2	141	661	581	71.77	63.09	67.15	44.75	39.33	41.87
	10	66.81	51	675	545	78.17	63.11	69.84	53.58	43.26	**47.87**
	Avg	**64.62**	**115**	**665**	**553**	**76.18**	**63.35**	**69.16**	**49.15**	**40.85**	**44.60**

Table 9: Results of applying SVM HMM to prepared dataset included discrete and three sources of word2vec embeddings

6.2 Semi-Supervised Learning

This section includes experimental results of employing semi-supervised learning techniques in research direction of open domain targeted sentiment classification. Next subsection show efficacy of applying label propagation model to open domain targeted sentiment classification. The other followed subsection shows experimental results that are provided when applying our proposed semi-supervised based solution.

6.2.1 Label Propagation

We developed an experiment to evaluate efficacy of applying label propagation model. We used only word2vec embeddings for training and testing label propagation model since this model uses only numeric data for finding nearest neighbours. The used datasets include only feature vectors that represent each word in the dataset (no information included for each tweet). We used "W2VPolyglotZhangBojanowski Norm" dataset for conducting this experiment since this data includes all pre-trained word2vec embeddings that are collected from the three resources. We selected different values for setting KNN (nearest neighbour) parameter. We changed as well the ratio of used labelled data of training set. We reported all results in Table 10 by using the same evolution metrics. We can notice clearly that this model is not suitable for solving our research problem because it does not consider the relation between words (tokens) in the same tweet.

Ratio %	**KNN**	**Pred #**	**NER**			**SA**		
			P	**R**	**F1**	**P**	**R**	**F1**
11	3	4205	4.68	60.8	8.7	0.48	6.17	0.88
51	3	4205	4.68	60.8	8.7	0.48	6.17	0.88
31	81	47	72.34	10.49	18.33	57.45	8.33	14.56
51	81	47	72.34	10.49	18.33	57.45	8.33	14.56
71	81	47	72.34	10.49	18.33	57.45	8.33	14.56
31	150	39	87.18	10.49	18.73	69.23	8.33	14.88
51	150	39	87.18	10.49	18.73	69.23	8.33	14.88
31	200	39	**87.18**	**10.49**	**18.73**	**69.23**	**8.33**	**14.88**
31,51	250,300	0	0	0	0	0	0	0

Table 10: Summary of best result when applying SVM HMM to the prepared datasets

6.2.2 Semi-Supervised Learning with Dynamic Generation of Feature Attributes

We developed an experiment to evaluate our proposed semi-supervised based solution. We use all results reported by research work [Li, 2017] to make our comparison and illustrate efficiency of our proposed solution. We changed ratio of labelled data into 25%, 50%, and 75% of training data. At each selected ratio of labelled data we applied both supervised SVM HMM and our proposed semi-supervised based model. We reported results of these both supervised and semi-supervised models to make the comparison easier and clarify the improvement in performance at each ratio of labelled data. With each ratio of labelled data we run as well optimization method for finding the optimum value of *C* parameter by finding lowest value of "zero/one-error" (Err). We changed value of *C* parameter from 1 into 550 with increase step equals 10. We applied this proposed model to all folds of DW2VPolyglotZhangBojanowskiNor dataset. All results provided by using both English and Spanish are reported in Table 11.

Lang	Model	Ratio	NER			SA		
			P	**R**	**F1**	**P**	**R**	**F1**
Eng	Supervised	25	64.65	48.57	55.08	45.76	34.19	38.85
	Semi-Supervised		64.20	50.18	55.84	45.92	35.77	39.86
	Supervised	50	66.46	51.92	58.21	47.86	37.31	41.88
	Semi-Supervised		66.43	53.46	59.13	48.34	38.81	42.97
	Supervised	75	68.93	51.86	59.15	50.56	38.00	43.36
	Semi-Supervised		**68.21**	**53.10**	**59.65**	**50.86**	**39.57**	**44.46**
Span	Supervised	25	68.86	61.12	64.72	40.04	35.53	37.63
	Semi-Supervised		67.48	62.97	65.05	39.90	37.21	38.45
	Supervised	50	73.73	61.18	66.84	45.42	37.66	41.15
	Semi-Supervised		71.48	64.02	67.39	44.15	39.51	41.61
	Supervised	75	74.66	62.34	67.93	47.06	39.27	42.80
	Semi-Supervised		**74.31**	**63.32**	**68.27**	**46.93**	**39.98**	**43.12**

Table 11: Average performance of applying semi-supervised learning with dynamic generation of feature attributes

The table includes also number of observed samples (obs) and number of samples (Pred) that are predicted correctly. The results include evaluations metrics of precision (P), recall (R), and F1-score (F1) for both name entity recognition (NER) and sentiment analysis (SA). The maximum values in this table are highlighted by using bold font.

6.3 Supervised Learning with Dynamic Generation of Feature Attributes

We developed experiments to evaluate efficacy of merging supervised SVM HMM with our proposed semi-supervised solution for generation feature attributes dynamically. We applied this combined supervised learning model to all folds of DW2VPolyglotZhangBojanowskiNor dataset. With each fold, we run optimization method for finding the optimum value of *C* parameter by finding lowest value of "zero/one-error" (Err). We changed value of *C* parameter from 1 into 550 with increase step equals 10. When clustering data that is used for generating feature attributes dynamically, we used a ratio of labelled data that is equal to 51% of training set.

All results achieved by applying these experiments to both English and Spanish data are reported in Table 12. The table includes also number of observed samples (obs) and number of samples (Pred) that are predicted correctly. The results include evaluations metrics of precision (P), recall (R), and F1-score (F1) for both name entity recognition (NER) and sentiment analysis (SA). The maximum values of accuracy and F1-score evaluated sentiment analysis are highlighted by using bold and underlined font. While the average values of all results provided when using all folds are highlighted by using only bold font. With each fold, we reported results of pure supervised learning when using the generated attributes does not increase performance.

Based on results reported in Table 12, we can note clearly that using three sources of word2vec embeddings decrease effect of missed words. After using these three sources most of words have at least one word2vec representation. We can note as well that Bojanowski word2vec embeddings outperforms the other two word2vec embeddings. While, concatenate all word2vec embeddings with discrete features provides the best results.

In general, using label propagation model provides bad results since it predicts each word (token) individually and does not consider the relations between tokens in the same tweet. Using a very small value of KNN parameter provides a fake result (the worst results) in which the number of predicted samples is greater than number of observed samples. Changing values of KNN parameter change results significantly, while changing values of labelled ratio does not make any change on the results.

To summarize our work in this research direction, we reported all results that are achieved by our proposed solution in comparison with previous related works. We reported the average of all values that are achieved by using all folds however using some specific folds provide better results. All main results that are achieved for open domain targeted sentiment classification are reported in Table 13. The table compares our proposed solutions with previous related works [Mitchell, 2013] (CRF-P, CRF-C, CRF-J), [Zhang, 2015] (NN-P, NN-C, NN-J), and [Li, 2017] (SS, SS(+w), SS(+P), SS(se)). The maximum achieved results are highlighted by using bold font. We can note clearly that SVM HMM model provides competitive results. Applying SVM HMM model by using discrete features with multiple word2vec embeddings outperforms all previous related works. We can notice as well that using some specific folds provide better results as shown in Table 9

Lang	**Fold**	**Err**	**C**	**Obs #**	**Pred #**	**NER**			**SA**		
						P	**R**	**F1**	**P**	**R**	**F1**
Eng	1	69.34	101	347	316	68.67	62.54	**65.46**	49.37	44.96	47.06
	2	70.75	21	324	254	72.83	57.1	64.01	55.91	43.83	49.13
	3	68.87	51	346	254	69.29	50.87	58.67	50.0	36.71	42.33
	4	73.11	51	318	268	65.67	55.35	60.07	48.51	40.88	44.37
	5	69.34	61	340	260	66.54	50.88	57.67	50.0	38.24	43.33
	6	68.87	31	319	271	67.9	57.68	62.37	49.82	42.32	45.76
	7	67.92	31	309	219	71.23	50.49	59.09	50.68	35.92	42.05
	8	69.34	21	320	229	76.42	54.69	63.75	60.26	43.13	**50.27**
	9	69.34	61	346	288	70.83	58.96	64.35	47.57	39.6	43.22
	10	69.81	31	319	225	71.11	50.16	58.82	52.89	37.3	43.75
	Avg	**69.67**	**46**	**329**	**258**	**70.05**	**54.87**	**61.43**	**51.50**	**40.29**	**45.13**
Span	1	64.87	81	677	576	76.04	64.7	69.91	50.17	42.69	46.13
	2	64.36	121	656	564	76.06	65.4	70.33	47.52	40.85	43.93
	3	62.42	151	676	571	74.61	63.02	68.32	48.34	40.83	44.27
	4	65.52	121	641	538	79.0	66.3	72.09	52.23	43.84	47.67
	5	64.58	111	669	604	79.3	71.6	**75.26**	51.49	46.49	**48.86**
	6	64.66	121	663	556	74.1	62.14	67.6	48.38	40.57	44.13
	7	65.44	141	651	533	76.17	62.37	68.58	47.28	38.71	42.57
	8	65.3	111	681	658	70.36	67.99	69.16	45.44	43.91	44.66
	9	62.2	141	661	665	66.62	67.02	66.82	42.11	42.36	42.23
	10	66.81	51	675	594	76.77	67.56	71.87	51.85	45.63	48.54
	Avg	**64.62**	**115**	**665**	**586**	**74.90**	**65.81**	**69.99**	**48.48**	**42.59**	**45.30**

Table 12: Results of applying supervised learning with dynamic generation of feature attributes

Model	English						Spanish					
	Entity Recognition			Sentiment Analysis			Entity Recognition			Sentiment Analysis		
	P.	R.	F1	P.	R.	F1	P.	R.	F1	P.	R.	F1
CRF-P	65.74	47.59	55.18	46.8	33.87	39.27	71.29	58.26	64.11	43.8	35.8	39.4
CRF-C	54.0	42.69	47.66	38.4	30.38	33.9	62.2	52.08	56.66	39.39	32.96	35.87
CRF-J	59.45	43.78	50.32	41.77	30.8	35.38	66.05	52.55	58.51	41.54	33.05	36.79
NN-P	60.69	51.63	55.67	43.71	37.12	40.06	70.77	62.0	65.76	46.55	40.57	43.04
NN-C	64.16	44.98	52.58	48.35	32.84	38.36	73.51	53.3	61.71	49.85	34.53	40.0
NN-J	61.47	49.28	54.59	44.62	35.84	39.67	71.32	61.11	65.74	46.67	39.99	43.02
SS	63.18	51.67	56.83	44.57	36.48	40.11	71.49	61.92	66.36	46.06	39.89	42.75
SS(+w)	66.35	56.59	61.08	47.3	40.36	43.55	73.13	64.34	68.45	47.14	41.48	44.13
SS(+P)	65.14	55.32	59.83	45.96	39.04	42.21	71.55	62.72	66.84	45.92	40.25	42.89
SS(se)	63.93	54.53	58.85	44.49	37.93	40.94	70.17	64.15	67.02	44.12	40.34	42.14
SVM HMM	70.08	54.38	61.16	50.88	39.42	44.36	76.18	63.35	69.16	49.15	40.85	44.60
Se-Su-DFG	68.21	53.10	59.65	50.86	39.57	44.46	74.31	63.32	68.27	46.93	39.98	43.12
Super-DFG	**70.05**	**54.87**	**61.43**	**51.50**	**40.29**	**45.13**	**74.90**	**65.81**	**69.99**	**48.48**	**42.59**	**45.30**

Table 13: Main results of open domain targeted sentiment classification

Using our proposed semi-supervised based solution (Se-Su-DFG) provides competitive results with less number of labelled data. The performance of this solution is close to accuracy of dominant previous related work. Thus, it is a good choice for using our proposed solution when there is a lack of labelled data or preparing it needs a costly process. Moreover, our proposed supervised based solution (Super-DFG) with dynamic generation of feature attributes outperforms all models that are proposed so far. To the best of our knowledge, these maximum results are not reported before with any related work for open domain targeted sentiment classification.

7 Conclusions and Future Work

In this work, we proposed supervised and semi-supervised based solutions with dynamic generation of feature attributes. Numerous empirical experiments are developed to show that our model outperforms all previous related works. To the best of our knowledge, this proposed solution achieved high performance which has not been reported before with any related work for open domain targeted sentiment classification.

Based on our experimental results, we can conclude that integrating discrete features with word2vec embeddings increases performance and decreases time complexity of open domain targeted sentiment classification when using CRF model in comparison with using neural network (NN) model. Additionally, adding word2vec embeddings as additional feature attribute will provide competitive accuracy with less implementation complexity in comparison with using additional feature layer in NN which is used by authors of [Zhang, 2015]. We can conclude also that applying SVM HMM model by using discrete features with multiple word2vec embeddings outperforms all previous related works. Moreover, using our proposed semi-supervised based solution provides competitive results with less number of labelled data. The performance of our solution is close to accuracy of dominant previous related work. Thus, it is a good choice for building models when there is a lack of labelled data while annotation process is a time-consuming.

This work can be extended in different directions. Since using word embeddings provides significant increase in accuracy, it is interesting to check efficiency of employing more forms of word embeddings in this research direction such as using global vectors for word representation (GloVe) [Pennington, 2014]. Additionally, it may be efficient to develop more new mechanisms for generating feature attributes automatically. Moreover, the future work should address effect of existence missed words that do not have word embeddings.

It is obvious that using SVM HMM is sensitive to the selected values of *C* parameter. Thus, it is important to compare performance of applying numerous optimization algorithms for finding optimum value of *C* parameter. Additionally, developing a new optimization solution for finding optimum value of *C* parameter may be an important direction. Moreover, it is interesting to evaluate efficiency of applying more sequence labelling models for improving performance of open domain targeted sentiment classification. We can also extend this research direction by checking efficacy of employing various deep learning techniques for improving accuracy of open domain targeted sentiment analysis. Merging semi-supervised with deep learning as well may improve performance and decrease the need for preparing labelled data. This combined technique may be a promising research direction since it was not addressed before.

Acknowledgements

The authors wish to acknowledge King Fahd University of Petroleum and Minerals (KFUPM) for providing the facilities to carry out this research. Spatial thanks to the Information Technology Center at KFUPM for providing high performance computing resources that have contributed to the research results reported within this paper. The authors would like as well to thank Dr. Yue Zhang and Mr. Duy-Tin Vo for answering questions and providing a pre-trained word2vec embeddings that are related to the dataset which has been used in this work.

References

[Abudalfa, 2013] Abudalfa, S., Mikki, M.: "K-Means Algorithm with a Novel Distance Measure"; Turkish Journal of Electrical Engineering & Computer Sciences, 21, 6 (2013) 1665-1684.

[Abudalfa, 2017] Abudalfa, S., Ahmed, M.: "Survey on Target Dependent Sentiment Analysis of Micro-Blogs in Social Media"; Proc. 9th IEEE GCC Conference & Exhibition (2017).

[Al-Rfou, 2013] Al-Rfou, R., Perozzi, B., Skiena, S.: "Polyglot: Distributed Word Representations for Multilingual NLP"; Proc. 17th Conference on Computational Natural Language Learning (2013) 183-192.

[Altun, 2003] Altun, Y., Tsochantaridis, I., Hofmann, T.: "Hidden Markov Support Vector Machines"; Proc. of the 20th International Conference on Machine Learning (ICML), Washington DC (2003) 3–10.

[Bojanowski, 2017] Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: “Enriching Word Vectors with Subword Information” Transactions of the Association for Computational Linguistics, 5 (2017) 135–146.

[Chamlertwat, 2012] Chamlertwat, W., Bhattarakosol, P., Rungkasiri, T., Haruechaiyasak, C.: “Discovering Consumer Insight from Twitter via Sentiment Analysis”; Journal of Universal Computer Science, 18, 8 (2012) 973-992.

[Chapelle, 2006] Chapelle, O., Schölkopf, B., Zien, A.: “Semi-Supervised Learning”; The MIT Press, London, England (2006).

[Hu, 2004] Hu, M., Liu, B.: “Mining and summarizing customer reviews”; Proc. of SIGKDD (2004) 168–177.

[Keerthi, 2007] Keerthi, S., Sundararajan, S.: “CRF versus SVM-Struct for Sequence Labeling”; Technical report, Yahoo! Research (2007).

[Klinger, 2013] Klinger, R., Cimiano, P.: “Bi-Directional Inter-Dependencies of Subjective Expressions and Targets and Their Value for a Joint Model”; in the Annual Meeting of the Association for Computational Linguistics (ACL) (2013).

[Li, 2017] Li, H., Lu, W.: “Learning Latent Sentiment Scopes for Entity-Level Sentiment Analysis”; Proc. of 31st AAAI Conference on Artificial Intelligence (AAAI) (2017) 3482-3489.

[Liu, 2015] Liu, B.: “SENTIMENT ANAYSIS: Mining Opinions, Sentiments, and Emotions”; Cambridge University Press (2015).

[Mikolov, 2013] Mikolov, T., Chen, K., Corrado, G., Dean, J.: “Efficient estimation of word representations in vector space”; Proc. of International Conference on Learning Representations (2013).

[Mitchell, 2013] Mitchell, M., Aguilar, J., Wilson, T., Durme, B.: “Open domain targeted sentiment”; Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP) (2013).

[Nguyen, 2007] Nguyen, N., Guo Y.: “Comparisons of Sequence Labeling Algorithms and Extensions”; Proc. of the 24th International Conference on Machine Learning (2007).

[Nguyen, 2007] Nguyen, N., Guo, Y. “Comparisons of Sequence Labeling Algorithms and Extensions”; Proc. of the 24th International Conference on Machine Learning (2007) 681–688.

[Nigam, 2006] Nigam, K., McCallum, A., Mitchell, T., “Semi-Supervised Text Classification Using EM”; The MIT Press, London, England, Ch. 3, (2006) 33-55.

[Parambath, 2014] Parambath, S., Usunier, N., Grandvalet, Y.: “Optimizing F-measures by Cost-Sensitive Classification”; in Neural Information Processing Systems (2014) 2123–2131.

[Pennington, 2014] Pennington, J., Socher, R. and Manning, C. D.: “GloVe: Global Vectors for Word Representation”; Proc. in Conference on Empirical Methods in Natural Language Processing (EMNLP) (2014) 1532–1543.

[Popescu, 2007] Popescu, A., Etzioni, O.: “Extracting product features and opinions from reviews”; In Natural language processing and text mining (2007).

[Qi, 2009] Qi, Y., Kuksa, P., Collobert, R., Sadamasa, K., Kavukcuoglu, K., Weston, J.: "Semi-Supervised Sequence Labeling with Self-Learned Features"; Proc. 9th IEEE International Conference on Data Mining (2009) 428–437.

[Ratinov, 2009] Ratinov, L., Roth, D.: "Design Challenges and Misconceptions in Named Entity Recognition"; Proc. of the 13th Conference on Computational Natural Language Learning, Stroudsburg, PA, (2009) 147-155.

[Wang, 2011] Wang, X., Wei, F., Liu, X., Zhou, M., Zhang, M.: “Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach”; In CIKM (2011) 1031–1040.

[Zhang, 2015] Zhang, M., Zhang, Y., Vo, D.: “Neural Networks for Open Domain Targeted Sentiment”; Proc. of the Conference on Empirical Methods in Natural Language Processing (2015) 612–621.