Exploiting Dependency Relations for Sentence Level Sentiment Classification using SVM

K Paramesha
Department of Computer Science & Engineering
Vidyavardhaka College of Engineering
Mysore, India-570002
Email: paramesha.k@vvce.ac.in

K C Ravishankar
Department of Computer Science & Engineering
Government Engineering College
Hassan, India-573201
Email: kcr@gechassan.ac.in

Abstract—In the sentiment analysis, finding the subjective clues itself is a challenging task. In this work, we propose a new approach, which employs Support Vector Machine (SVM) for classification, exploits the dependency relations in a dependency tree coupled with a large lexicon resource obtained from twitter to create a feature vector. The experiment shows a significant improvement over the baseline approaches and results are on par with existing methods in two-class classification.

Keywords—Dependency Relations, Feature Engineering, Senti-ment, NRC Hashtag Sentiment Lexicon.

I. Introduction

Sentiment analysis being a multifaceted problem [1] has a lot of economic stakes if the results are better for the desired objectives. As we have studied the literature in this area, there is one thing that is clearly understood and expressed in [2] that there are many open-ended problems at different strata. We think that there is a scope for alternative means using variety of information derived from Part-of-speech tagger, syntactic parser, Name Entity Recognition tagger etc. to develop a model which could be deployed in the real time environment and the fine tuning of the system over a time period and on running input data. In this process of designing a suitable model, several approaches such as based on lexicon, rule based, probability, vector model, NLP and machine learning have been put forward. In these models, feature engineering plays a vital role in making of working model. Deriving a discriminating feature set for a robust model is a challeng-ing and in-exhaustive task, yet every piece of information associated with the data if properly explored and exploited, could be valuable. In our proposed work, we have used the dependency information, which is a relation between words in a dependency tree for a sentence parsed by a syntactic parser and established that features derived from the dependency relation produces better results.

II. Related Work

Sentiment analysis is primarily a classification problem dealt at different levels. At the sentence level sentiment analysis, several methods have been proposed to show the improvement in the performance over the baseline methods. The work closest to ours using dependency relation components is found in [3] where the polarity of a combination of governor and dependent in the dependency relation is a score computed based on heuristic rules. The score is propagated on recursive application of the rules resulting in a final score of the sentence polarity. However, if there is an inversion in the polarity at a level, then subsequent scores are affected and eventually the final score. In contrast, we use the combinations of polarities of governor and dependent as vector fields.

Thinking on same lines as in [4] where features exploited from the dependency tree gives information on interaction between words proved to have yielded better results, so we have witnessed a significant appreciation in the results.

With regard to the data-set used in [5] for testing after the model was trained on a large corpus, the results obtained on testing data showed improvements over lexicon methods. The model is designed to perform fine grain sentiment analysis while trained on coarse level sentiment. Due to lack of large annotated set at sentence level, they chose polarized documents based on star-rating for training the model. Compare to our method using the same data-set where we use dependency relation , in [5] features at word level were exploited to predict sentence label. Since we tested our model on same data-set, we discuss the results in the section VI.

III. Methodology

In our proposed approach, we chose vector model over conditional random fields(CRF) because of two reasons. (1) Since the CRF had already been applied onto the data-set and also given the fact that in CRF, it is understandably difficult to estimate parameters with initial priority values [4], we think that there is a scope for alternative means using feature engineering. (2) SVM is very efficient if the vectors are computed correctly for the belonging classes and that's where the dependency tree comes in handy in generating vectors. Our intuitions are illustrated with examples later in the section.

In our approach, there are two phases: (1) Deriving the feature vector from dependency tree using lexicon resource.
(2) Classification of the vector into positive( )/negative( ) polarity using SVM.

A. Feature vector generation

On analyzing dependency trees, The polarities of a pair of nodes (governor and dependent) connected by an edge representing one of the many typed dependency relations such as nn, amod, advmod, advcl can appear in one of the combinations (a,b,c & d), nsubj, aux, cop, det, prep, neg

978-1-4799-6085-9/15/$31.00©2015 IEEE

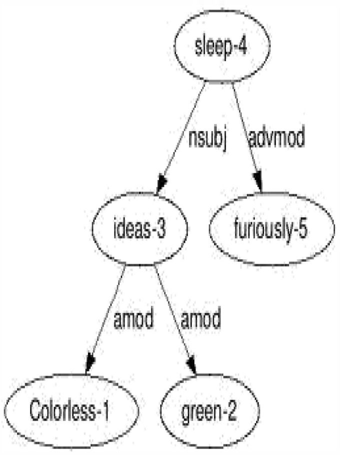


Fig. 1: Dependency tree showing relation between words.

can appear in one of the combinations (e & f) and pobj can appear in one of the combinations (g & h).
For e.g. the combinations for advmod, nsubj and pobj dependency relation are shown below(!represents neutral/don't care).

advmod advmod
(a) ! (b) !
advmod advmod
(c) ! (d) !
nsubj nsubj
(e) ! (f) !
pobj pobj
(g) ! (h) !

For sentence level sentiment analysis, first we parse the sentence using STANFORD PARSER to yield the dependency tree, from which, a feature vector of the sentence is generated. An example of a dependency tree is shown in Fig. 1.

The fields of a feature vector spans all the combinations of each of the selected dependency relations which captures the sentiment words. The polarities of sentiment words mapping into the combination are determined using lexicon resources discussed in section IV. The general format of a feature vector where the fields corresponding to the different combinations of different relations is shown below.

advmod nsubj
! ; ! ; ; ! ; ! ; ; +

z }| { z }| {
In the above vector format, the fields representing all the four combinations under advmod contain the respective count of occurrences of the combination in the dependency tree.
For e.g. in a vector h2; 1; 0; 0; 0; 0; ; 0; 0; 0; 0; 1i the first field value 2 is the count corresponds to the first combination
(a) under advmod in the dependency tree.

The notion of having generated vectors based on the dependency relations is that the feature vector generated for a positive polarity sentence will be distinct from the feature vector for the negative polarity sentence. The fact that the difference in the vector is induced due to the structural differences and also the polarities of words contributing towards the polarities of sentences. To illustrate our views, consider the following cases.

1) Case1: The dependency trees in Fig. 2(a) and Fig. 2(b) are for two opposite sentences resulting in structurally dif-ferent. The feature vector fields under the dependency rela-tion neg in Fig. 2(a) register 0 whereas for the Fig. 2(b) it registers the count 1, thereby generating a discriminating feature vectors. Since the vector fields corresponding to the different combinations of selected dependency relations, the model inherently captures the structural features.

2) Case2: The dependency trees in Fig. 3(a) and Fig. 3(b) are for two opposite sentences resulting in structurally same, but in the feature vector, the field corresponding to the combi-nation ! under the dependency relation dobj in Fig. 3(a) register count 1 whereas for the Fig. 3(b) it registers the count 1 in the feature vector for the different field corresponding to the combination ! under the same dependency relation dobj, thereby still generating a discriminating feature vectors. In this case, even if there is no discrimination with respect to structure, the model is still enabled to incorporate the discriminating features based on the polarities of words forming the different combinations under different dependency relations.

B. Classification using SVM

The feature vectors thus obtained for sentence level and document level are fed to the linear SVM classifier to perform binary classification at sentence and document level separately. We perform 10 fold cross-validation process to evaluate the performance of our model.

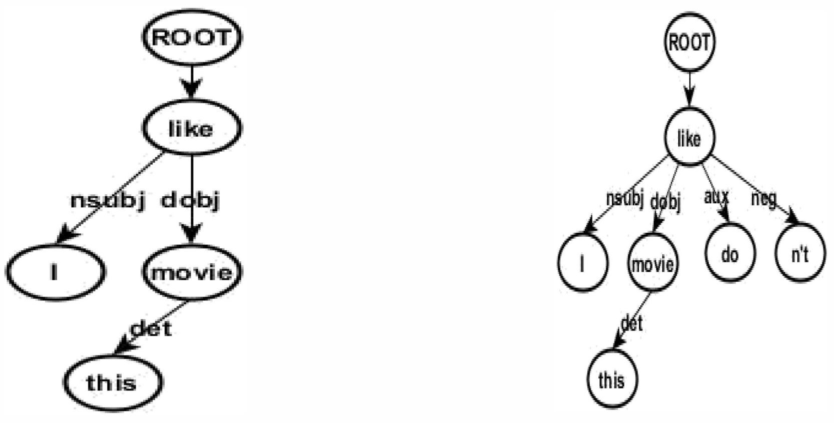
IV. Lexicon Databases

As with the most of the sentiment analysis techniques, we employ two freely available lexicon resources namely SentiWordNet and NRC Hashtag Sentiment Lexicon derived automatically from tweeter. NRC lexicon is based on Pointwise Mutual Information(PMI) of words co-occurring with positive and negative sentiment words. It was first introduced in [6] by Mohammad,Svetlana and Zhu and saw a remarkable results. We tried our experiments on uni-gram lexicon which gives polarity of word in the value range -5 to +5.

V. Experiments

We tested the proposed model on the data-set used in our previous work [7]. We performed the following baseline experiments on both sentence and document level. The document vector has been generated by summation of sentence level vectors in the document.

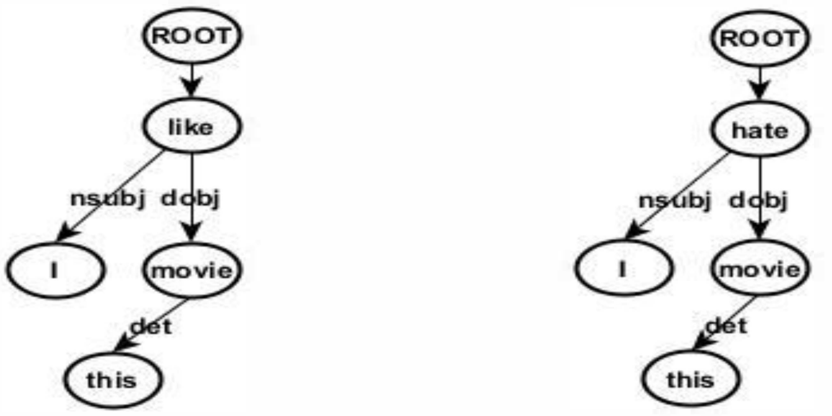
SemEval : For each word in a sentence, the PMI value from the NRC lexicon decides the polarity of the word. Based on voting, the polarity of the sentence is computed.



((a)) I like this movie.

((b)) I don't like this movie.

Fig. 2: Dependency trees for two opposite statements showing different structure.



((a)) I like this movie.

((b)) I hate this movie.

Fig. 3: Dependency trees for two opposite statements showing same structure.

SentiWN : The method is same as the SemEval but the scores are computed using the SentiWordNet.

SVMonSemEval : The SVM classifier is employed on feature vector of data for binary classification. For each sentence in a document, feature vector is generated using dependency parse tree and NRC lexicon as discussed in section III-A.

SVMonSentiWN : The method is same as the SVMonSe-mEval but the polarities of sentiment words are computed using the SentiWordNet.

VI. Results

The results of our model is furnished in Table I. In comparison with the results stated in [5] for sentence level in binary classification, our model results are indeed better at sentence level and on par with other methods at document level. Further, comparing with the results of our previous work in [7], which uses bag-of-words, dependency relation once again proved to be more promising.

TABLE I: Accuracy at sentence and document level

Method	Sentence	Document
SentiWN	55.4	63.0
SemEval	60.3	64.5
SVMonSentiWN	65.4	74.3
SVMonSemEval	70.2	80.5

VII. Conclusion and Future work

As we have observed, the syntactic information is indeed provide a fillip to the performance as compared to the mere bag-of-words. Although our feature vector doesn't encompass diverse information such as contextual information, it can still be able to perform better at both sentence and document level. We are also cognizant of the fact that enriched lexicon can make the difference in the performance. In the case of NRC lexicon, which already had established many records, obviously turned out to be a significant source for sentiment lexicon. In our future work, we seek to enrich the feature set from diverse sources to enhance the existing performance and test the model on several data-sets.

References

[1] B. Liu, "Sentiment analysis: A multi-faceted problem," IEEE Intelligent Systems, vol. 25, no. 3, pp. 76–80, 2010.

[2] K. Paramesha and K. C. Ravishankar, "A perspective on sentiment analysis," in Proceedings of ERCICA 2014 - Emerging Research in Com-puting, Information, Communication and Applications, vol. 1. NMIT, Bengaluru, India: Elsevier, August 2014, pp. 412–418.

[3] L. K.-W. Tan, J.-C. Na, Y.-L. Theng, and K. Chang, "Sentence-level sentiment polarity classification using a linguistic approach," in Digital Libraries: For Cultural Heritage, Knowledge Dissemination, and Future Creation. Springer, 2011, pp. 77–87.

[4] T. Nakagawa, K. Inui, and S. Kurohashi, "Dependency tree-based senti-ment 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. Association for Computational Linguistics, 2010, pp. 786–794.

[5] O. Tackstr¨om¨ and R. McDonald, "Discovering fine-grained sentiment with latent variable structured prediction models," in Advances in Infor-mation Retrieval. Springer, 2011, pp. 368–374.

[6] S. Mohammad, S. Kiritchenko, and X. Zhu, "Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets," in Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013), Atlanta, Georgia, USA, June 2013.

[7] K. Paramesha and K. C. Ravishankar, "Optimization of cross domain sentiment analysis using sentiwordnet," arXiv preprint arXiv:1401.3230, 2013.