Semantic evaluation of natural language in tweets for classification

Alfonso Alvarenga
School of Computing
KAIST
alfonso82@kaist.ac.kr
20154607

Henrik Holm
School of Computing
KAIST
henholm@kaist.ac.kr
20176337

Þorsteinn Daði Gunnarsson
School of Computing
KAIST
gunnarsson@kaist.ac.kr
20176488

abstract

I. Introduction

II. Problem definition

We decided to work on the SemEval task of Sentiment Analysis on Twitter, which was the task number 4 in the SemEval of 2016. The main purpose of this project is to analyze a database of tweets in order to establish a sentiment classification for each individual tweet. The tweets to be analyzed may either belong to a certain topic or not. Different sentiment analysis approaches will be used. For instance, in the case of analyzing tweets belonging to a certain topic, the task is to decide whether the tweets express a positive or a negative view about that topic. Going into further details, three subtasks will be worked on. As described on the SemEval 2016 webpage (http://alt.qcri.org/semeval2016/task4/), they are the following[1]:

- **Subtask A: Message Polarity Classification:** Given a tweet, predict whether the tweet is of positive, negative, or neutral sentiment. This is a single label multi-class classification task where each tweet should be assigned as POSITIVE, NEUTRAL or NEGATIVE.
- **Subtask B: Tweet classification according to a two-point scale:** Given a tweet known to be about a given topic, classify whether the tweet conveys a positive or a negative sentiment towards the topic. This is similar to task A except there are only two classes making this a binary classification problem.
- **Subtask C: Tweet classification according to a five-point scale:** Given a tweet known to be about a given topic, estimate the sentiment conveyed by the tweet towards the topic on a five-point scale. Each tweet should be assigned one of five ordered classes from HIGHLYPOSITIVE to HIGHELYNEGATIVE where possible classes are C={HIGHLYPOSITIVE, POSITIVE, NEUTRAL, NEGATIVE, HIGHLYNEGATIVE} This is an ordinal classification task, meaning mistakes are not all equally weighted.

Although each subtask is different and require different approaches, they share common points that we can exploit for similar preprocessing and feature selection processes.

III. Related work

Many SemEval-2016 tasks' solutions have been proposed [2]. As we are only working on task 4, we made a research on those papers dedicated particularly to this task. In this section we made a brief summary on those we found interesting and from which we believe we can extract important techniques and ideas to propose our own solution. Thus, this section drives not on language processing for semantic evaluation *per se*, but on Sentiment Analysis in Twitter[1] approaches.

In the most general ways, there are two dominant families of solutions for this problem: machine learning approaches (both as supervised and unsupervised learning), and lexicon-based approach (with a dictionary of words) [3]. CUFE [4], for example, makes use of a Neural Network (NN) to learn sentiment from sentences. The core of their network is a Gated Recurrent Unit (GRU) layer, more computational efficient than Convolutional Neural Network (CNN) models (Lai et al., 2015), and it can capture long semantic patterns without tuning the model parameter, according to the authors. In contrast, [5] uses a CNN with an input that represents the text as a concatentation of its word embeddings. They also applies a normalization step that includes normalizing urls and mentions.

It is worth mentioning that Swiss-cheese [6] method achieved a good performance in SemEval2016 [2], and it is build in a CNN model. For the network's input, each word is associated in Swiss-cheese to a vector representation, which consists in a d-dimensional vector. A sentence (or tweet) is represented by the concatenation of the representations of its n constituent words. Here, all the feature extraction step if left to the CNN layers.

Similar to Swiss-cheese, is SENSEI-LIF [7], with a CNN architecture that relies on word embeddings as word representation as well as sentiment polarity lexicon features, concatenated to the word representation. One of the purpose of its CNN is to extract sentence-level features in order to model global evidence in the tweet. Word embeddings in SENSEI-LIF includes lexical, sentiment, and part-of-speech embeddings.

Other approaches as TwiSE [8] have a more complicated step of feature modeling before the learning step, including a

5 step feature extraction process and a feature representation and transformation step. The authors of TwiSE experimented with several families of classifiers such as linear models, maximum-margin models, nearest neighbours approaches and trees, evaluating their performance in the data provided by the organisers of SemEval2016 [1]. They found out that the two most competitive models were Logistic Regression from the family of linear models, and Support Vector Machines (SVMs) from the family of maximum margin models.

The method proposed in [9], Tweester, have a predominant feature extraction process, based on the assumption that semantic similarity implies affective similarity. Tweester feature extraction process includes tokenization, Word2vec word representation, statistic extraction, and other additional features. Its final model is a fusion of Topic modeling, CNN, Word2Vec system, and Webis.

IV. Proposed ideas

Though each task requires classification of tweets they are fundamentally different so one approach will not work for every one. However, they do have some things in common so parts of the solution could apply to one or more tasks.

A. Tokenization

The tokenization process can be shared between all three tasks. For this it could be useful to take into consideration *hashtags* since we are working with tweets and they can incorporate some sentimental value. For simplicity this means tokenizing by word would be preferable or if other methods, like n-grams, are used extracting *hashtags* as a second set of features could prove valuable. A drawback of tokenizing only by words is that tweets can include spelling errors and abbreviations. This should be taken into account to minimize the feature space and noise of the data. Emojis are popular for conveying emotion and are a very useful feature for these tasks and the presence of one should definitely be accounted for.

The Python NLTK library[10][11] includes a nice Tweet tokenizer that is splits tweets into words, emojis and hashtags. This would be a good starting point for tokenization.

B. Feature selection

Feature selection is an important step for both accuracy and efficiency. For better performance choosing features that hold sentimental value is rather important. This means weighting words based on appearances is not the optimal stragegy. For task B and C, that both include classification given a topic, we have to keep in mind that each feature can have different sentimental value based on the topic. For example the word *horrifying* would normally be considered negative but in the case of a horror movie it could be considered positive.

C. Classification

This is where the subtasks differ the most. Each one needs a slightly different approach when it comes to classification and what methods are used. Based on the results on [4] [7] [6], out classification method will be NN model that includes the feature modeling ideas of [8] and [9]. The purpose of this is mostly to test how a fully implemented CNN model that is already sufficient to extract its own features in a sentence level [6] would perform when some level of feature extraction is already given. We believe that even the minimal feature representation can increase performance.

Other sub-task related ideas that we are considering includes:

1) Subtask A: Since subtask A include three different classes a single classifier is not enough. We plan to use a support vector machine (SVM) with a one-vs-rest strategy. That is we will train three classifiers, one for each class and the one with the best match wins. This idea is supported by the results found in [8].

2) Subtask B: Subtask B is a binary classification problem so a single SVM will do. A solution as simple as this one is supported by [8], as in subtask A.

3) Subtask C: Subtask C includes ordinal classification with five classes. This means using the same approach as in subtask A would be plausible but that means the classifier does not take into account or use for its advantage the order of the classes. We do not have an idea how to solve this at the moment.

V. Intermediate Results

For the moment, we don't have any intermediate result to report.

VI. Discussion

References

[1] P. Nakov, A. Ritter, S. Rosenthal, V. Stoyanov, and F. Sebastiani, "SemEval-2016 task 4: Sentiment analysis in Twitter," in *Proceedings of the 10th International Workshop on Semantic Evaluation*, ser. SemEval '16. San Diego, California: Association for Computational Linguistics, June 2016.

[2] S. Bethard, D. M. Cer, M. Carpuat, D. Jurgens, P. Nakov, and T. Zesch, Eds., *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*. The Association for Computer Linguistics, 2016.

[3] V. A. Kharde and S. Sonawane, "Sentiment analysis of twitter data : A survey of techniques," *CoRR*, vol. abs/1601.06971, 2016. [Online]. Available: http://arxiv.org/abs/1601.06971

[4] M. Nabil, A. Atyia, and M. Aly, "Cufe at semeval-2016 task 4: A gated recurrent model for sentiment classification," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, 2016, pp. 52–57.

[5] S. Ruder, P. Ghaffari, and J. G. Breslin, "INSIGHT-1 at semeval-2016 task 4: Convolutional neural networks for sentiment classification and quantification," *CoRR*, vol. abs/1609.02746, 2016.

[6] J. Deriu, M. Gonzenbach, F. Uzdilli, A. Lucchi, V. D. Luca, and M. Jaggi, "Swisscheese at semeval-2016 task 4: Sentiment classification using an ensemble of convolutional neural networks with distant supervision," in *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, 2016, pp. 1124–1128.

[7] M. Rouvier and B. Favre, "SENSEI-LIF at semeval-2016 task 4: Polarity embedding fusion for robust sentiment analysis," in *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, 2016, pp. 202–208.

[8] G. Balikas and M. Amini, "Twise at semeval-2016 task 4: Twitter sentiment classification," *CoRR*, vol. abs/1606.04351, 2016.

[9] E. Palogiannidi, A. Kolovou, F. Christopoulou, F. Kokkinos, E. Iosif, N. Malandrakis, H. Papageorgiou, S. S. Narayanan, and A. Potamianos, "Tweester at semeval-2016 task 4: Sentiment analysis in twitter using semantic-affective model adaptation," in *SemEval@NAACL-HLT*, 2016.

[10] E. Loper and S. Bird, "Nltk: The natural language toolkit," in *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ser. ETMTNLP '02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 63–70.

[11] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*, 1st ed. O'Reilly Media, Inc., 2009.