Aspect-based sentiment analysis

Avtorja: Julijan Jug, Jaka Jenko

I. Introduction

Sentiment analysis is a part of natural language processing, that tries to identify and extract opinions within a given text for example in comments, social media posts, articles, etc. Most of these texts are written in a very unstructured way. The benefit of sentiment classification is that it can extract more structured sentiment data that can be easily analyzed. Most commonly sentiment is classified into three classes positive, neutral, and negative.

In this paper, we focused on classifying aspect-based sentiment, which is the sentiment towards a certain aspect that the text is about. This presents its own set of problems when trying to represent the data in such a way that it captures the relevant aspect-based information. We set out to implement a classification model that is capable of classifying sentiment and test it on a data set of 837 documents in Slovene, they contain 31,419 entities with coreferences.

II. Related work

Several papers addressed similar problems of classifying sentiment of short texts or sentiment towards certain entities in those texts.

A. Ding et. al.

In their paper, Entity-Level Sentiment Analysis of Issue Comment[1] authors obtained a database of issue comments from several GitHub projects and annotated their sentiment into three classes. They used a fairly standard preprocessing step that included stop-word removal, tokenization, stemming, and vectorization using TF-IDF and Doc2Vec method. For the prediction, they used Random Forest, Bagging, SVM, Naive-Bayes, Gradient Boosting models. They achieved a classification accuracy of 68%.

B. Sweeney et. al

In a paper titled Multi-entity sentiment analysis using entity-level feature extraction and word embeddings approach[2], authors used a word embeddings approach to sentiment classification. Word embeddings enable the representation of semantics. This paper proposed a sentiment lexicon-based technique to appoint a total score that indicates the polarity of opinion concerning an entity. The used the proposed method on a database ob 1.5 mio annotated tweets. They classified tweets into only two classes and achieved an accuracy of 71%.

C. Biyani et. al

In a paper titled Entity-Specific Sentiment Classification of Yahoo News Comments[3], authors aimed to classify sentiment towards entities in Yahoo News comments. Their approach includes two steps. In the first step authors extracted the relevant entities and in the second step classify the sentiment towards them. In the experiment, they used context extraction and proposed the use of non-lexical features that are an improvement over previously used lexicon-based features and bag-of-words. They achieved the F-score of 67%.

D. Go et. al.

In a paper titled Twitter Sentiment Classification using Distant Supervision[4], authors described the preprocessing steps in preparation of twitter data and used machine learning algorithms including Naive Bayes, Maximum Entropy, and SVM. For the feature extractors, they used unigrams, bigrams, unigrams and bigrams, and unigrams with part of speech tags. Emoticons were mapped to their respective text counterparts and addressed in the feature extraction part. They achieved a classification accuracy of over 80%.

E. Jiang et. al.

In paper Target-dependent Twitter Sentiment Classification[5], authors addressed the problem of sentiment in tweets. They proposed two methods of improving target dependant sentiment classification by increasing context information. The incorporated some target dependant features and took into consideration other related tweets. They showed that the addition of these features greatly improves the performance of systems for sentiment classification.

III. Data preprocessing

For the experiment we used SentiCoref 1.0[6] data set, which contains 837 documents in Slovene. The documents have annotated named entities (persons, organizations, and locations) and their coreferences. It includes 31.419 tagged named entities and 14.572 coreference chains. Before we could start working with data we had to preprocess it.

For each entity in data we extracted:

- File ID
- Entity ID
- Entity type
- Entities (all words representing the entity)
- Sentiment
- Words before (5 words before the entity)
- Words before sentiments (vector of sentiments for each of those 5 words)
- Sentences (whole sentences in which entity appears)

In "words before", "words before sentiments" and "sentences" we have also removed all stop-words and lemmatize the words.

For removing stop words we used the nltk python library, for Slovenian word lemmatization we used lemmagen.lemmatizer and to get word sentiments, we used annotated data from Slovene sentiment lexicon KSS 1.1[7].

IV. Baseline model

Firstly we created a classification model that uses an input in a format similar to document-term matrix. That means that every column represents a certain word and rows represent groups of 5 words preceding the entity in the original text. The size of that array is 14.905 different words and 48.528 entities. This matrix is very large and sparse and thus very difficult to use for learning classification models. Because of this, we have reduced the size of the matrix by removing the columns where the word repeats less than 40 times and all rows with less than 2 words. Doing this we reduced the data set to 754 words and 12.665 lines.

	KNN	NB	RF	MLP
3 classes (words)	0.594551	0.256612	0.637583	0.622581
2 classes (words)	0.650217	0.523884	0.652191	0.673904

	KNN	LR	RF	MLP
2 classes (sentiments)	0.74825	0.75453	0.74965	0.75209
3 classes (sentiments)	0.74651	0.74755	0.73500	0.74651

Classes were also reduced from 5 to 3 classes (positive, neutral and negative) with distribution of:

- Positive - 0.158902
- Neutral - 0.621594 (majority class)
- Negative - 0.219502

And from 3 to 2 classes (neutral, other) with distribution of:

- Neutral - 0.626924 (majority class)
- Other - 0.373075

We tried the learning the model with KNN, Naive Bayes, random forest and multi-level perceptron neural network. This models achieved the following classification accuracy:

The Naive Bayes algorithm performed badly. With an accuracy score of less than the majority class in both cases. The accuracy of models was better when we predicted only 2 classes, but it was still only 5% better than the majority class at best.

For the second baseline model we used the Slovene sentiment lexicon KSS 1.1[7] and mapped the five words before each entity with the corresponding sentiment. This process produced vectors of different sizes due to the early end of sentences and removal of stop words. The training data consisted of three attributes: number of positive, neutral, and negative words in each entity occurrence. The distribution of classes looks like this:

- Positive - 0.126
- Neutral - 0.741 (majority class)
- Negative - 0.131

We predicted the sentiment using KNN, Naive Bayes, Logistic regression and multi-level perceptron and achieved the following classification accuracy:

On 2 classes multilevel perceptron performed the best with an accuracy of 0.752. And on 3 classes Logistic regression performed the best with 0.748 accuracy. The results are not particularly good when compared to the majority class percentage. We tried to upsample the minority classes to try and mitigate the effect of majority class on the model learning, but it didn't show any significant improvement.

For the third classification model, we used Elmo embeddings. We computed the Elmo embeddings for five words before the target entity and included them in the training data. With this approach, we achieved an accuracy of 0.7475.

V. Conclusion

For future work, we intend to extend and improve the use of embeddings and maybe combine some of the approaches used in the first three baseline models.

References

[1] J. Ding, H. Sun, X. Wang, and X. Liu, "Entity-level sentiment analysis of issue comments," 06 2018, pp. 7–13.

[2] C. Sweeney and D. Padmanabhan, "Multi-entity sentiment analysis using entity-level feature extraction and word embeddings approach," in *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*. Varna, Bulgaria: INCOMA Ltd., Sep. 2017, pp. 733–740. [Online]. Available: https://doi.org/10.26615/978-954-452-049-6_094

[3] P. Biyani, C. Caragea, and N. Bhamidipati, "Entity-specific sentiment classification of yahoo news comments," 2015.

[4] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N project report, Stanford*, vol. 1, no. 12, p. 2009, 2009.

[5] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, "Target-dependent twitter sentiment classification," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, ser. HLT '11. USA: Association for Computational Linguistics, 2011, p. 151–160.

[6] S. Žitnik, "Slovene corpus for aspect-based sentiment analysis - SentiCoref 1.0," 2019, slovenian language resource repository CLARIN.SI. [Online]. Available: http://hdl.handle.net/11356/1285

[7] K. Kadunc and M. Robnik-Šikonja, "Slovene sentiment lexicon KSS 1.1," 2017, slovenian language resource repository CLARIN.SI. [Online]. Available: http://hdl.handle.net/11356/1097