Aspect-based sentiment analysis

Miha Bizjak, Anže Gregorc, Rok Grmek
University of Ljubljana
Faculty for computer and information science
Večna pot 113, SI-1000 Ljubljana
mb9232@student.uni-lj.si, ag9497@student.uni-lj.si, rg6954@student.uni-lj.si

Abstract

TODO

1 Introduction

Online news, forums and social media are a place for everyone to read and write articles and posts across various domains. People can also leave comments and giving their opinion and express their feelings about the topics. That leads to a huge amount of text content. That is probably why natural language analysis is currently a hot topic around the world. We wanted to extract useful information out of large amount of text data.Since we have no time for reading all the words that are written nowadays, we hope to build a good computer program to do that for us. In this project we chose to do aspect-based sentiment analysis. Our task is to get the subjective information from text material that refer to a entity with the use of natural language processing and other methods. An entity is considered as a person, organization or a location and can be represented multiple times in one document or a sentence and there could be more entities in one document.

For the given task, we decided to test multiple approaches and develop different models for predicting the sentiment for each entity. We will first define some really simple models as a starting point, and then we will try to derive some more complex models. All of them will be targeting the Slovene language, and we will evaluate each of the models on the Slovene corpus for aspect-based sentiment analysis - SentiCoref 1.0 (Žitnik, 2019).

2 Related work

The main challenge of entity-based analysis is how to find words that describe the entity and identify if contributes to positive or negative sentiment to a given entity. A lot of related work tried to predict sentiment of the whole document. But in many cases, a text can describe the polarity of more entities. That is why we suggest that sentiment analysis is done on entity level. Since our task is more specific we focused more on methods that identify entities.

In the paper (Ding et al., 2018) they developed an entity-based sentiment analysis SentiSW and tested it on issue comments from GitHub. SentiSW can classify issue comments into <*sentiment, entity*> tuples. They evaluate the entity recognition by manually annotation and it achieves 75% accuracy. The main pipeline of this tool is preprocess (words removal, words replacing, stem), feature vectorize (TF-IDF, Doc2vec), classifier (random forest, bagging and other supervised machine learning methods) and entity recognition (rule-based method).

The use of word embeddings provide powerful methods for semantic understanding without the need of creating large amounts of annotated test data. The paper (Sweeney and Padmanabhan, 2017) enhanced the word embeddings approach with the deployment of a sentiment lexicon-based technique to appoint a total score that indicates the polarity of opinion in relation to a particular entity. They associate a given entity with the words describing it and extracting the associated sentiment to try to infer if the text is positive or negative in relation to the entity.

As stated in the paper (Song et al., 2019), a lot of the existing approaches are modelling context and target words with RNNs and attention. This paper addresses the issues with RNNs and proposes an Attentional Encoder Network for modeling the semantic interactions between target and context words. The paper also addresses the label unreliability issue. The proposed model with the use of pre-trained BERT embedding achieved state-of-the-art results while still being a relatively lightweight model.

Another successful approach that utilizes the BERT model is described in the paper (Sun et al., 2019). In this paper, the authors tried a few methods, where they were generating an auxiliary sentence for each prediction. They basically converted the aspect-based sentiment analysis problem into a sentence-pair classification task.

3 Methods

3.1 Dataset

The SentiCoref 1.0 dataset contains 837 documents with annotations of named entities (31,419 entities in total) and sentiment annotations for each entity. For each entity, a sentiment value from 1 to 5 is assigned. The distribution of sentiment labels in the dataset is shown in Table 1.

Sentiment label	Entity count
1 - Very negative	30
2 - Negative	1801
3 - Neutral	10869
4 - Positive	1705
5 - Very positive	24

Table 1: Entity sentiment distribution in the SentiCoref 1.0 dataset.

3.2 Models

TODO

3.2.1 Random model

TODO

- Randomly assigns sentiment from 1 to 5 to each entity.
- No real value.
- Developed along with the evaluation toolbox only for testing.
- Also serves as a model that should be evaluated with the lowest possible score.

3.2.2 Majority model

TODO

- Assigns the neutral sentiment (3) to all entities.
- Should produce a decent score because of the distribution of the sentiment classes.
- Will serve for a reference score - complex models should not be performing worse than this simple majority model.

3.2.3 Lexicon Features model

TODO

- The model uses a random forest classifier with the following features:
  - number of positive words up to 5 words before/after each entity occurrence,
  - number of negative words up to 5 words before/after each entity occurrence,
  - number of positive words for which the current entity is the closest,
  - number of negative words for which the current entity is the closest,
  - number of positive words in sentence,
  - number of negative words in sentence,
  - number of positive words in the text,
  - number of negative words in the text,
  - number of different entities in the text,
  - number of occurrences of the entity.
- The optimal parameters for the model are chosen using grid-search with 5-fold cross-validation.
- Sentence splitting is done using the pre-trained Punkt sentence tokenizer (Kiss and Strunk, 2006) for Slovene provided with the `nltk` Python package (Bird et al., 2009).

3.2.4 Further ideas

TODO (note: following ideas will be used for implementing a few more models)

- Current features in the Lex. Feat. model depend mainly on the positive/negative words in the neighborhood of the entity. Instead of looking at the neighborhood, we could use a dependency parser and observe sentiment of the most related words (not necessary in the neighbourhood).
- Instead of constructing handcrafted features, we could use BERT model for feature extraction. Those features depend on the context of each word, so we could simply use feature representation of each entity word occurrence and its sentiment as a learning sample

for some classifier. With the trained classifier, we could predict sentiment for each occurrence of the entity and calculate the average sentiment for the entity.

- Test the effect of using different classifiers (re-implement a single model with different classifiers)
- Combine multiple feat. representations into a single model (normalize and concatenate feature vectors from different models and let the classifier use/learn most important features from all models).

3.3 Evaluation

TODO

- 2/3 train, 1/3 test split
- Implemented measures: Accuracy, Precision, Recall, F1 score.

4 Results

TODO

Table 2 lists the results of our models on the test dataset.

Model	Accuracy	F1-score
Random model	0.203	0.128
Majority model	0.751	0.172
Lexicon Features	0.756	0.197

Table 2: Model results on the test dataset.

5 Discussion

TODO

6 References

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Jin Ding, Hailong Sun, Xu Wang, and Xudong Liu. 2018. Entity-level sentiment analysis of issue comments. In *Proceedings of the 3rd International Workshop on Emotion Awareness in Software Engineering*, pages 7–13.

Tibor Kiss and Jan Strunk. 2006. Unsupervised multilingual sentence boundary detection. *Computational linguistics*, 32(4):485–525.

Youwei Song, Jiahai Wang, Tao Jiang, Zhiyue Liu, and Yanghui Rao. 2019. Attentional encoder network for targeted sentiment classification.

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 380–385, Minneapolis, Minnesota. Association for Computational Linguistics.

Colm Sweeney and Deepak Padmanabhan. 2017. Multi-entity sentiment analysis using entity-level feature extraction and word embeddings approach. In *RANLP*, pages 733–740.

Slavko Žitnik. 2019. Slovene corpus for aspect-based sentiment analysis - SentiCoref 1.0. Slovenian language resource repository CLARIN.SI.