Aspect-based sentiment analysis on SentiCoref 1.0 dataset

Julijan Jug, Jaka Jenko
University of Ljubljana
Faculty of computer and imformation science
Večna pot 113, SI-1000 Ljubljana
jj6706@student.uni-lj.si, jj3580@student.uni-lj.si

Abstract

The paper analyses different classification models and feature sets for aspect-based sentiment prediction. Previous research has shown that aspect-based sentiment prediction is a very hard problem, which is magnified due to the peculiarities of the Slovene language.

The experiments were conducted using data from SentiCoref 1.0. The data set was prepossessed by extracting entities, corresponding sentiments, and the surrounding words. the words were lemmatized and cleared of stop words. Additional data with more examples of how the entities are used was also added with the help of SentiNews corpus and Slovene sentiment lexicon of positive and negative words.

Baseline results show a slight improvement over the majority class with 0.65 classification accuracy and the F-score of 0.64. Better results were obtained with the use of extended feature set from SentiNews and Slovene Sentiment Lexicon the use of the ELMo training model with 0.75 classification accuracy and the F-score of 0.70 on 2 class classification. On 3 class classifications, the best model achieved an F-score of 0.68 and a classification accuracy of 0.74.

1 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 and used in other practical application purposes. 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 extracted from the SentiCoref corpus. SentiCoref includes 837 documents in Slovene, they contain 31,419 entities with coreferences.

2 Related work

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

2.1 Ding et. al.

In their paper, Entity-Level Sentiment Analysis of Issue Comment(Ding et al., 2018) 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%.

2.2 Sweeney et. al

In a paper titled Multi-entity sentiment analysis using entity-level feature extraction and word embeddings approach(Sweeney and Padmanabhan, 2017), 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%.

2.3 Biyani et. al

In a paper titled Entity-Specific Sentiment Classification of Yahoo News Comments(Biyani et al., 2015), 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%.

2.4 Go et. al.

In a paper titled Twitter Sentiment Classification using Distant Supervision(Go et al., 2009), 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%.

2.5 Jiang et. al.

In paper Target-dependent Twitter Sentiment Classification(Jiang et al., 2011), 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.

3 Data preprocessing

For the experiment we used SentiCoref 1.0 (Žitnik, 2019) 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 stopwords 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(Kadunc and Robnik-Šikonja, 2017).

4 Baseline model

Firstly we created a classification model that uses an input in a format similar to the 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.

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

- Positive - 15.89%
- Neutral - 62.15%
- Negative - 21.95%

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

- Neutral - 62.69%
- Other - 37.30%

We tried learning the model with Random forest and Logistic regression. These models achieved the following classification accuracy and F-score as presented in Table 1.

	RF	LR
2 classes	F: 0.56 CA: 0.65	**F: 0.64** CA: 0.65
3 classes	F: 0.51 CA: 0.63	**F: 0.58** CA: 0.63

Table 1: Classification accuracy and F-scores of random forest and logistic regression on base document-term matrix

The accuracy of models is higher when predicting only 2 classes, which is to be expected but is still at best only 3% better than the majority class.

We constructed another baseline model that predicts only the majority sentiment class. We used a somewhat different data set from the first baseline model. In this data set we treated each entity as a sample and joined the contexts of its every coreference in a combined context. The context corresponds to five words before the coreference occurrence. This resulted in fewer samples and a different distribution for 3 class data set:

- Positive - 12.65%
- Neutral - 74.18%
- Negative - 13.16%

And for 2 class data set distribution:

- Neutral - 74.18%
- Other - 25.81%

The Majority classifier achieved an F-score of 0.66 on 3 class problem and an F-score of 0.63 on 2 class problem.

5 Methods

In this experiment we used several methods for classification of sentiment. We tried different approaches to building models as well as using different sets of features. These features included vector mapping approaches like ELMo vectors and also features extracted from other sources like SentiNews corpus and Slovene sentiment lexicon.

We constructed several sets of features and used different combinations of these features in the model learning process. Features include:

- ELMo embeddings
- SentiNews sentiments
- Slovene sentiment lexicon features

5.1 ELMo embeddings

ELMo is a deep contextualized word representation method that models the complex characteristics of word use and how these uses vary across linguistic contexts (Peters et al., 2018).

We used ELMo embeddings as a way to represent the context of the entities. The goal was to capture how the entities are being used in context and hopefully capture the information about the expressed sentiment towards that entity. The problem is how to define the context. We decided to use five words preceding each entity occurrence and its coreferences as a sufficient context.

Our preprocessed data structure includes five words that precede the entity occurrences in the document. This effectively gives us the surrounding area of every entity occurrences. These words were lemmatized and include no stop words. Because we want to represent each entity with only one ELMo vector we decided to merge all of its occurrences. This resulted in only one context string for every entity. For the computation of the ELMo embeddings we used a pre-trained language model trained on the 1 Billion Word Benchmark included in the TensorFlow library. The input into the ELMo model was split into smaller batches for performance benefits. This process produces a single ELMo embedding for every entity.

5.2 SentiNews

SentiNews is a Slovenian news corpus with manually annotated sentiment (Bučar, 2017). It consists of 10,427 documents from the Slovenian news portals 24ur, Dnevnik, Finance, Rtvslo, and Žurnal24. It includes sentiments annotations on three levels of granularity, i.e. on the document, paragraph, and sentence level. The texts were annotated by 6 annotators using the five-level scale (1

– very negative, 2 – negative, 3 – neutral, 4 – positive, and 5 – very positive). We focused primarily on the average sentiment score of these 6 annotators, the standard deviation of the sentiment annotations, and the most frequent sentiment.

For the construction of features using this corpus, we first build three indexes for the document, paragraph, and sentence level sentiments for easier inquiries. We tokenized the contents of every document in the corpus and extracted only entities that represent persons, organizations, or locations. For categorizing entities we used nltk library and its tokenize feature. We then extracted the average sentiment, standard deviation, and the overall sentiment (neutral, positive, negative) on the respective granularity level. We then filtered out the entities that do not also appear in the SentiCoref data set and saved the rest in an index in the form of a Numpy object. An example of an index row is presented in Table 2.

entity_id	avg_sen	std_sen	mode_sen
9966-29	2.4375	0.088375	neutral

Table 2: Example of an row in the document level index that includes entity id, average sentiment, standard deviation, and mode sentiment.

We ended up with 9 features. Figure 1 presents the correlation between the features and target variable sentiment. We can observe that target sentiment does not correlate highly with any of the features. There is some correlation between other features, the highest correlation of 0.59 is between average sentence-level sentiment and average paragraph-level sentiment. The standard deviation in document sentiment has the highest correlation to the target variable with 0.11.

5.3 Slovene sentiment lexicon

Slovene opinion lexicon KSS is a lexicon of positive and negative words typical for the Slovenian language (Kadunc and Robnik-Šikonja, 2017). We used all three versions of the lexicon that include lemmatized words and their sentiment. We assigned sentiment to the words in the entities' surroundings, that is five words before the occurrence, and constructed a vector of sentiments. We then counted the number of negative, positive, and neutral words in the sentiment vector which gave us the number of positive, negative, and neutral words.

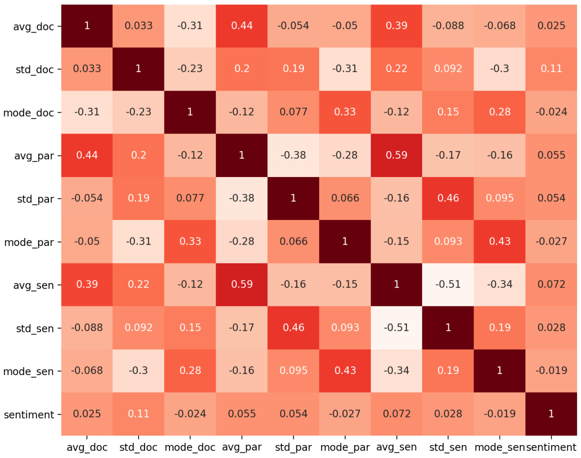

Figure 1: Correlation between features from SentiNews and target varaible sentiment.

Figure 2 presents the correlation between the features and target variable sentiment. The number of negative words has the highest correlation of 0.2 with the target variable.

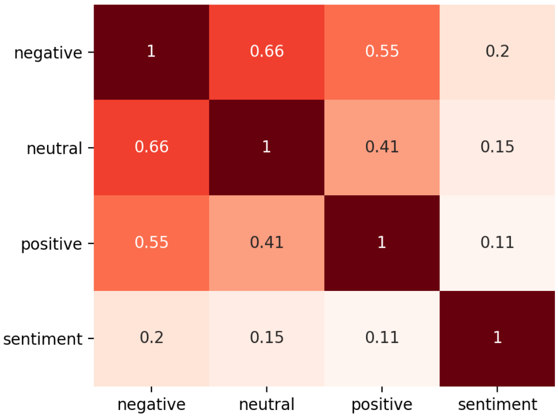

Figure 2: Correlation between features from Slovene sentiment lexicon and target varaible sentiment.

6 Results

The experiment consisted of trying different feature sets for learning the Random forest and Logistic regression models. All models were trained on the same train test data set split using an 80:20 ratio. The models were tested on 2 and 3 class classification problem.

For the first model we used only the ELMo embeddings and achieved results listed in Table 3.

Table 4 includes classification accuracy and F-score on data set that includes features from SentiNews, Slovene sentiment lexicon, and ELMo embeddings.

	RF	LR
2 classes	F: 0.64 CA: 0.69	F: **0.65** CA: 0.66
3 classes	F: 0.63 CA: 0.70	F: **0.64** CA: 0.67

Table 3: F-scores and classification accuracy using only ELMo embeddings.

	RF	LR
2 classes	F: 0.68 CA: 0.73	F: **0.69** CA: 0.75
3 classes	F: 0.65 CA: 0.73	F: **0.67** CA: 0.74

Table 4: F-scores and accuracy using ELMo, sentinews and kss features for 2 and 3 class problem.

Table 5 includes classification accuracy and F-score of models on data set that includes features from SentiNews, Slovene sentiment lexicon, and ELMo embeddings. This data set was upsampled in an attempt to improve the results. The minority classes were upsampled to represent half of all samples.

	RF	LR
2 classes	F: **0.64** CA: 0.66	F: 0.63 CA: 0.62
3 classes	F: 0.63 CA: 0.70	F: **0.64** CA: 0.64

Table 5: F-scores on upsampled data set using ELMo, SentiNews and KSS features.

Table 6 includes classification accuracy and F-score of models on data set that includes only features from SentiNews and Slovene sentiment lexicon.

	RF	LR
2 classes	F: **0.70** CA: 0.73	F: 0.68 CA: 0.75
3 classes	F: **0.68** CA: 0.73	F: 0.66 CA: 0.74

Table 6: F-scores on data set using only SentiNews and KSS features.

7 Conclusion

Our goal was to predict the sentiments of entities from the SentiCoref 1.0 (Žitnik, 2019) data set and improve on the baseline model performance. First we preprocessed the SentiCoref data, extracted entities, sentiments, and words surrounding them and the coreferences. We created two simple base models with majority classifier and predictions based on a document-term matrix to get baseline results. Then we implemented ELMo embedding to contextualize words and extended the data set with the use of SentiNews data set (Bučar, 2017) and Slovene sentiment lexicon (Kadunc and Robnik-Šikonja, 2017) features. The best achieved results are a classification accuracy of 0.74 and an F-score of 0.68 for 3 class problem.

The biggest problem we encountered was the unbalanced data set. The data consisted of 74% of neutral sentiment examples and not so many polarised (negative and positive) examples. We tried to mitigate this unbalance by upsampling the minority classes but that did not achieve the desired results. F-score using the upsampled data set dropped to 0.64.

Models using only ELMo embeddings for sentiment prediction, did not turn out very good. They achieved an F-score of 0.64, which is one of the lower scores. The problem may lie in the way we defined the ELMo vector and how we used entity coreference surroundings in its construction. For a better understanding we should analyze the use of different context definitions in the construction of ELMo embeddings.

By using only SentiNews and KSS features the F-score improved to 0.68. If we look at the classification accuracy we see that it dropped in comparison to the majority classifier although it has a higher F-score. This is due to the fact that the model has a higher recall of positive and negative sentiments and consequently also false-positive neutral sentiments. This feature set outperformed the models that included ELMO, SentiNews, and KSS features. This indicated that ELMo embeddings did not capture the context in the way we anticipated in the beginning.

When looking at the 2 class classification the results are slightly better with the highest F-score being 0.70 and classification accuracy 0.75. Better results are to be expected because of the nature of classification complexity.

The final model barely improved the baseline models performance but is better at finding the positive and negative sentiments while still being able to categorize neutral cases. For future work, we intend to extend and improve the use of embeddings and combine some of the approaches used in the first three baseline models. We also intended to use BERT embeddings but decided to leave it for future research. The one thing that we didn't do and it could improve the classification of 3 classes, was to first predict only neutral and none neutral cases and after that classify the none neutral cases into positive and negative.

References

Prakhar Biyani, Cornelia Caragea, and Narayan Bhamidipati. 2015. Entity-specific sentiment classification of yahoo news comments.

Jože Bučar. 2017. Manually sentiment annotated slovenian news corpus SentiNews 1.0. Slovenian language resource repository CLARIN.SI.

Jin Ding, Hailong Sun, Xu Wang, and Xudong Liu. 2018. Entity-level sentiment analysis of issue comments. pages 7–13.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009.

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, page 151–160, USA. Association for Computational Linguistics.

Klemen Kadunc and Marko Robnik-Šikonja. 2017. Slovene sentiment lexicon KSS 1.1. Slovenian language resource repository CLARIN.SI.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations.

Colm Sweeney and Deepak Padmanabhan. 2017. 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*, pages 733–740, Varna, Bulgaria. INCOMA Ltd.

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