Project for class NLP: Aspect-based sentiment analysis

Dina Sarajlić[1] and **Sanja Stojanoska**[2] and **Vanda Antolović**[3]
Faculty of Computer and Information Science, University of Ljubljana
Email: [1]ds0267@student.uni-lj.si, [2]ss3151@student.uni-lj.si, [3]va2351@student.uni-lj.si

Abstract

One of the most recent opinion mining research is discovering polarity of specific entities contained in a given text. At the beginning aspect-based sentiment analysis was primarily used for reviews analysis where the aspects of an entity are clearly defined and the goal is to get detailed feedback from a user, but its usage has expanded due to the day-to-day increase of online contents. This paper describes machine learning model, based on the lexical knowledge of the content, intended to determine polarity of entities from Slovenian news articles.

1 Introduction

Sentiment analysis on entity level is an important research in the field of Natural language processing (NLP). Sentiment analysis has become popular topic in the past few years, at the beginning used for determining polarity of a given document or text, but it has expanded since. As the Web content got enriched with many product and service reviews, tweets and comments, there was an increased need for fine-grained sentiment analysis, also called Aspect-based sentiment analysis (ABSA), to get better insight of a user's opinion. ABSA determines the polarity of each aspect, identifies sentiment's orientation to positive, negative or neutral.

This paper proposes ABSA on news articles. The goal is to build a hybrid approach of lexical structure and machine learning to determine the polarity of each given entity in a particular news text. It is organised as follows, section 2 gives insight on Related work that have given us inspiration and provided useful information on Sentiment analysis in general, section 3 explains the Methods we have implemented, with section 4 presenting gotten results. Ultimately in section 5 we conclude our findings.

2 Related work

In general there are two main approaches for sentiment classification; knowledge based approach and machine learning approach.

Knowledge based approach uses predefined lexicons of opinion words labeled as: positive, negative or neutral. In this case the sentiment is determined by comparing the text of interest with the pre-defined entry in the lexicon. Machine learning approach involves training a sentiment classifier. Many related researches use: Naive Bayes (NB), Support Vector Machines (SVM) and Maximum Entropy.

The (Sweeney and Padmanabhan, 2017) research aims to investigate how entities and their descriptors can be used to identify the sentiment of tweets in relation to one entity or many entities if more than one entity exists. The important novelty here is that they treat the tweets differently which have only one entity and the ones with more. In addition, many-entity tweets are analyzed in a way that particular descriptor words are extracted as features and their sentiment is identified using SentiWordNet lexicon while one-entity tweets are processed using the Word2Vec algorithm.

(Ding et al., 2018) designed entity-level sentiment analysis tool SentiSW based on four modules: preprocessing, feature vectorization, sentiment classification and entity recognition. Preprocessing step aims to reduce noise words and uses stemming techniques. Vectorization module transfers bag of words (BOW) into vectors with the help of TF-IDF and Doc2Vec. Sentiment classification classifies comments into neutral, positive and negative. Last module takes only subjective sentiment sentences and recognizes the entity as 'Person' or 'Project' towards which the sentiment refers to. Their goal is to determine emotion on each entity written in a GitHub issue comment.

The designed system outputs a tuple of (sentiment, entity) if the comment is subjective or 'neutral statement' if the text is objective.

(Biyani et al., 2015) addresses entity-specific sentiment classification of comments written on Yahoo News. It is formulated as two-stage binary classification. First, filtering the relevant entities. Second, classification of relevant entities as positive or negative. The approach follows three phases: context extraction, feature generation, sentiment classification. Context extraction connects each entity with its context in the given text. In case a sentence does not contain entities, its context belongs to all other entities. On the other hand, if a sentence contains more than one entity, only phrases are taken as context related with each of the entities. Feature generation uses knowledge based approach to find interesting features. They noticed that entity of type person is more likely to be polar, compared to an entity of non-person type, which is a useful fact for the first-step classification. Moreover, they use algorithm for calculating sentiment score and create features like SentiPos and SentiNeg.

Sentiment classification is done by using machine learning algorithm, more specifically Logistic Regression for the neutral-polar classification and Naive Bayes for positive-negative classification. Model evaluation resulted into 67% as F1 measure for the neutral-polar classification and 70% F1 score for the positive-negative.

3 Methods

3.1 Dataset

Given dataset (Žitnik, 2019) contains 14,572 entities with a 5-level sentiment annotation, starting with 1 as very negative to 5 as very positive, from which 75% entities are marked as neutral, positive and negative annotations have similar sizes around 12% each, while the rest 1% is consisted of entities which are very positive and very negative. According to this data distribution, we can conclude that the dataset is imbalanced and therefore some preprocessing has to be done.

3.2 Preprocessing

The specified problem is simplified by joining classes of very positive and very negative to positive and negative class respectfully. The idea for such simplification is due to low representation of the extreme classes in the datataset. Simple reasoning for such class imbalance is that the news are unbiased source of information and usually do not contain many polarised word descriptions.

Preprocessing step is an important starting point for emphasising only aspects which are relevant and helpful to determine the sentiment of each entity. Therefore we left out numerical data as well as special characters. Using LemmaProcessor by Stanza (Qi et al., 2020), we extracted lemmas for each word. Moreover, with the features provided by POSProcessor by Stanza(Qi et al., 2020) we got a part of speech tag for each word and with the help of DepparseProcessor(Qi et al., 2020) we were able to find paths into the grammar dependency tree.

Since many lexical properties are meaningful for successful sentiment analysis, we used several lexicons (Bučar, 2017b; Kadunc and Robnik-Šikonja, 2017) to get polarity score for the opinion words within given article. Usually stopwords do not have some informative content. Being compared to the opinion words from the lexicon, most of the stopwords if do have a polarity, their score is still neutral, so the aricles are analyzed without stopwords.

3.3 Feature extraction

An article is transformed in such a way that it is split into different subsets for each named entity. All of the subsets contain the main entity and its coreferences. A main entity is the entity to which all of the coreferences into the subset belong to. In most cases it is personal name or a noun. The idea is to create feature vector for each entity by exploiting the properties of the words contained in that entity subset. Using Byani et al (Biyani et al., 2015) as a baseline we included similar features for polar-neutral classification. In addition:

- **isPerson** checks whether an entity subset has a word with 'PER' named-entity tag.
- **isSubject** checks if a word in the subset has a relation 'nsubj' meaning that this word occurs to be subject within some sentence.
- **isObj** feature describes if there is a word that has a dependency relation 'obj' meaning that this word happened to be a direct object in the article.
- **hasClues** checks if the entity subset contains a polarized word.

- **isNegated** marks whether the word was preceded by a negation verb.

- **hasDescriptors**, descriptor words are verbs, adjectives and adverbs, and this feature has a value of 1 if a word in the subset has a POS tag 'VERB'. If this is not the case than it is checked if in the window of +/-3 words, the ocurences of an entity, contain some descriptor words. This is done considering the fact that descriptor words are more likely to express some emotion.

- **contextPolarity** contains useful information for the polarity of an entity. To calculate this feature there are many approaches that can be used due to the wide range of possibilities to be considered as a context of a word.
  First, using a window around an entity we calculated the polarity of its neighborhood.
  Other solution was to split the article by sentences and assign each sentence to the context of entity, while discarding sentences without entities.
  Moreover, dependency parser builds a grammar tree from each sentence given in a document. Using this ability and having entities marked, we search for a shortest path connecting an entity with an opinion word whose lemma is contained in the lexicon and its polarity is different than zero. In this way, each entity occurrence has the polarity of its pair opinion word. Overall contextual polarity of an entity is gathered from the polarities of its coreferences.
  In addition, using the dependency relations in a sentence, we experimented by searching the grammar tree for a specific type of relation as 'amod', 'advmod' and 'obj' and assigned polarity to the entity that has the right connection with the opinion word. The results of the above mentioned experiments are discussed below in 4.
  We have also tried determining concordances polarity which should have helped with defining polarity of an entity more precisely. Concordances are appearances of a word and its surrounding environment (in our case we had used a window of 7 words before and after the main entity), as a main entity were only taken words that have a POS tag of noun, pronoun or adjective. Polarities of each word in concordance sentences was extracted directly from the lexicon (cla), and as main entity's final polarity we placed one that is most common in its concordance sentences.

- **polarDocument** is a feature generated using sentiNews (Bučar, 2017a) dataset and it has value 1 if the document is polar, otherwise 0.

At the end of this step each entity has its feature vector which will be used for classification.

3.4 Classification

Essential approach for the aspect-based sentiment analysis is to create binary classification which will distinguish between two classes polar and neutral, as well as positive and negative.

3.4.1 Polar-neutral classification

First, we started by transforming the multiclass dataset into binary classes: neutral entities (given with polar score 3) and polar entities (all other polarity scores) This resulted into a dataset containing 10,727 neutral and 3,387 polar samples. Due to the highly imbalanced dataset, we used the following approaches to improve the overall F1 score for polar-neutral classification:

- **using classifiers over total dataset**
- **undersampling and oversampling**
- **neutral vs all**

Algorithms used for all three approaches that produced currently best results are: Logistic Regression, Random Forest Classifier and Support Vector Classification. In addition, we experimented with Gradient boosting, Naive Bayes and K Neighbors Classifier.

Using classifiers over the whole dataset - done simply over the imbalanced dataset with Logistic regression, Random Forest Classifier and SVC given a parameter `class_weight="balanced"`. These classifiers are close in performance and their results can be compared in the table 1 below. Without weighted balancing of the classes' parameters the classifiers mostly produced low scores.

Undersampling and oversampling - by using NearMiss, SMOTE and RandomUnderSampler over data samples. They have produced results as gotten with adding above mentioned parameter into the classifiers, hence this is not a preferenced way of solving the imbalance of a dataset.

Usually these workaround approaches will rarely score well on an unknown dataset because the nature of data has different distribution.

Neutral vs all - we have tried dividing the neutral set of data into four smaller datasets. These neutral samples from dataset were divided as groups 40-40-10-10, while the polar set was kept original from very negative to very positive. Each degree of polarity was put into a set with one of the neutral groups (positive and negative each with one of the 40%, very positive and very negative each with one of the 10%). Ultimately this idea proved to not give higher results for classifying polar data.

3.4.2 Positive-negative classification

For the positive-negative classification, the neutral data was eliminated from the dataset. The remaining samples are divided into two classes one containing positive and very positive while the other negative and very negative. This was a balanced set with 1,748 negative samples and 1,639 positive. Having sets with comparable sizes, now we have freedom to use classifiers other than ones which provide class balancing. The evaluation of this step classification and the results is explained in the section below and can be seen in table 2.

As a second approach for classification, we performed two-step classification. First step was consisted of applying specific classifier in neutral-polar classification. After obtaining classification, we extracted feature values from test-set which were 'ground-truth' polar and predicted also as polar. Those feature values were taken as model for second-step classification. In the second-step we performed positive-negative classification by applying range of classifier as usual.

We tested this approach on 2.csv set which contains data with very low amount of pre-processing and dataset 6.csv for which we obtained best results in positive-negative classification with first approach. For both datasets, we used in first-step for polar-neutral classification a classifier with highest f-score which is NaiveBayes for 2.csv and RandomForrest for 6.csv. Further, on newly obtained sets of feature values for second-step, the results were following:

- For 2.csv dataset, results were much better in whole range of classifiers; NaiveBayes, RandomForrest, LogisticRegression,SVC and KNeighboursClassifier. We noticed major improvement in all evaluation measures; precision, recall, f-score and accuracy. The highest f-score weighted average for both classes with previous approach was 0.53 (NaiveBayes) and with two-step classification 0.70 (LogisticRegression). However, this was expected since there was not sufficient preprocessing towards positive-negative classification with previous approach.
- For 6.csv dataset, results for both classes(positive and negative) were up to some coefficient more balanced, but not significant difference was noticed. Compared by f-score weighted average, in previous approach NaiveBayes and LogisticRegression were more successful by very low percentage and in two-step classification RandomForrest, SVC and KNeighboursClassifier were more successful.

We can conclude that two-step classification should be performed when there is quite unprepocessed dataset. Also, we must take into consideration that size of dataset for second-step classification was very short but the results satisfying. That favors importance of a good train dataset.

4 Results and discussion

Both classification settings are evaluated following the same baseline. Based on some features manipulation we created different models trying to achieve best results for the minority class. As described above in the section's 3 **contextPolarity** feature, there are many experiments regarding the polarity context of an entity, therefore the models are: neighbourhood polarity (**N**) - takes neighbourhood words around each entity mentions as its context, sentence polarity (**S**) - takes the sentence as a context in which the entity is found (and no other entities), shortest path polarity assignment (**SP**) - finds shortest path between an entity mention and opinion word in the same sentence, and weighted relations (**WR**) polarity - which investigates different type of relations in the dependency tree and assigns weight to the ones containing opinion word. Moreover this model is combined with the window context and uses the lexicon from (Kadunc and Robnik-Šikonja, 2017). Unfortunately, context concordances function did not prove to give good results for our model, hence it had not been included in table 1. Moreover, we discarded verb mentions from the entity specific subset and used document polarity as a feature which slightly increased the results.

Through different models the Random Forest classifier and Support Vector classifiers have similar results and have good average performance for the neutral-polar classification. The table 1 shows average results sorted by the best results with respect to the deficit class. Lower performance of Naive Bayes classifier can be result of the lack of class balance. We tried removing specific features from the model to see how they influence the received results. Results differentiated the most with removing docPol feature, with F1 score lowering by at most 5%. What appeared to influence as well are the isSubject and isObject features, which removed at the same time lower the F1 score by most for 3%. Removing other features either minimally influenced the score or had not at all. Changing the size of the training set, between 20% and 30%, does not greatly influence the score of the classifiers either.

Table 1: Table shows several models and their performance for the polar-neutral classification

model	Pr	Re.	F-1
WR + Support vector	**0.73**	**0.68**	**0.70**
S + Random forest	**0.73**	**0.68**	**0.70**
S + Support vector	0.73	0.67	0.69
N + Random forest	0.72	0.67	0.68
SP + Logistic regression	0.72	0.65	0.67

Positive-negative classification has been evaluated using the same models as in the previous step while we tried two other classifiers: Neural network classifier and Gradient boosting classifier. The best results for this step were achieved by Naive Bayes classifier with the weighted relations model (WR). Other models are not far behind as it can be seen in the table 2.

As it can be seen in the tables, the weighted relations model gives satisfactory results. This is achieved by specific walks in the dependency tree for searching an opinion-important grammatical relations (such as: amod, advmod or obj) and assigning weights to the connections.

Table 2: Table shows several models and their performance for the positive-negative classification

model	Pr	Re.	F-1
WR + Naive Bayes	**0.76**	**0.76**	**0.76**
WR + Gradient boosting	**0.76**	**0.76**	**0.76**
WR + Linear regression	0.75	0.74	0.74
N + Support vector	0.74	0.72	0.72
N + Linear regression	0.74	0.72	0.71

5 Conclusion

In this paper, we studied the problem of identifying the sentiment of entities from news articles. The proposed model gathers lexical features which describe the grammatical structure and semantic context within the text to build a model which determines sentiment class for each entity. Working with articles covering wide variety of topics is challenging, mostly for extracting the context of entities. Usually articles are written in a well-explanatory way providing a lot of information which makes it a difficult task to map an entity to its context. Larger contexts hides the polarity of the rare descriptor words and small context does not provide enough sentiment. Future work in this field may focus on discovering important relations between entities and descriptor words by creating unique pairs with weighted grammatical distances.

References

Reldianno – text annotation service for processing slovenian, croatian and serbian. https://www.clarin.si/info/k-centre/web-services-documentation/. Accessed: 2020-03-30.

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

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

Jože Bučar. 2017b. Slovene sentiment lexicon JOB 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. In *Proceedings of the 3rd International Workshop on Emotion Awareness in Software Engineering*, page 7–13. Association for Computing Machinery.

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

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082*.

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.