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

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.

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 Implemented baseline

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

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. It should be noted that before mapping words to their lemmas we kept track of the negated version of the verb 'to be', because due to the lemmatization the negation will be lost. 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 Depparse-Processor(Qi et al., 2020) we got the dependency relation between that word and its head in the dependency tree. Since many lexical properties are meaningful for successful sentiment analysis, we used a lexicon (Bučar, 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 score it is still neutral. Due to this reasoning stopwords are removed from the dataset.

3.3 Feature extraction

An article is transformed in such a way that it is split into different subsets for each mentioned 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. Descriptor words are verbs, adjectives and adverbs
- **hasDescriptors** 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 main entity contains 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 extract context for each entity, the main article is transformed with respect to the sentences of the article. Once the sentences are extracted and their entities are known each entity gets a sentence to be in his context if it is the only entity there. It is assumed that entities which will not have context to be neutral because in

most cases it happens to be locations or organizations. A further improvement of this feature is to split multi-entity sentences and include their phrases to the context.

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

3.4 Classification

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

First, we started by transforming our multi-class dataset into binary classes, we have done so by exchanging the Polarity column values in our dataset, Neutral marked values were changed with 0 and others (positive, negative, very positive, very negative) with 1 denoting the Polar data. This resulted into our dataset containing 10,727 Neutral and 3,387 Polar data.

Due to the highly imbalanced dataset, we used the following approaches to improve the overall F1 score for polar-neutral classification:

- using classifiers over the whole dataset
- undersampling and oversampling
- neutral vs all

Learning algorithms used for all three approaches that produced the currently best results for us are: Logistic Regression, Random Forest Classifier and Support Vector Classification. We have also tested with Naive Bayes and K Neighbors Classifier, both have not provided sufficiently good results.

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"`, with random forest also having set parameter `n_estimators=200`. Best result for prediction of neutral and polar was the precision score of 0.73, recall score of 0.65, giving the weighted average F1 score of the model 0.68, as can be seen in table 1. Without these parameters the classifiers mostly produced low scores.

Undersampling and oversampling - by using NearMiss, SMOTE and RandomUnderSampler over data. They have produced results as gotten with adding above mentioned parameters into the classifiers, hence this is not the preferenced way of solving the imbalance of our dataset. Usually these workaround approaches will rarely score well on an unknown dataset because the nature of data has different distribution.

Table 1: Table shows several models and average their performance

model	Pr	Re.	F-1
Random Forest	0.73	0.65	0.68
Logistic regression	0.73	0.64	0.67
Support vector	0.74	0.65	0.67

Neutral vs all - we have tried dividing the neutral set of data into four smaller datasets. These datasets 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 predicting polar data,

To evaluate success for the positive-negative classification, the neutral data was eliminated from the dataset. We have again transformed the labels into two classes one containing positive and very positive marked with 1, hence the negative and very negative were marked with 0. This was a balanced set with 1,748 negative samples and 1,639 positive. Using different mentioned algorithms we have gotten close predictions for these two, best was with Logistic Regression with F1 score 0.62 for Negative, 0.52 for Positive, with weighted average of 0.57.

4 Future directions

As for now, the most of the improvements in future tasks will be implementing additional preprocessing steps and adjusting current ones. Main thoughts are to increase context reliability and to make analysis more robust to metaphorical language and irony. We consider following improvements:

- handling words under quotation marks and assigning inverted polarity before removing all special characters
- best improvement would be to replace acronyms and slang phrases with definition of the acronyms; for this purpose we would

check if there are any web-sources to obtain the lists of commonly utilized acronyms. If that would not be possible, we will just extract dot and observe abbreviation as separate entity

- recognizing irony pattern and applying it by taking context into consideration
- using different dataset as lexicon (i.e. (Kadunc and Robnik-Šikonja, 2017))
- include more negation words if needed

Currently, for context extraction we tried windows of 3 tokens before and/or after a word in sentence depending on word location and tokens polarity which didn't give expected results. Further improvements can be based on assigning corresponding weights to the words in an entity environment based on their polarity and context. Option is to exploit the features from dependency parsing and take as a context the path connected to the entity with the opinion word. Also, the major improvement would be to analyze clauses in sentence and extract all entities with corresponding feature vectors. Features would be extracted also based on interraction with other clauses in sentence. The most common example would be to take into consideration adverbs implying opposite polarity such as 'ampak'.

As we do not have at this point satisfactory quality in neutral-polar classification, as previously mentioned, we have done the positive-negative classification as separate task omitting all neutral entities. Improvement would be to observe it as second step of neutral-polar classification in a way to proceed only with entities classified as polar.

The idea was also to perform very positive/very negative classification. Due to deficient amount of data (cca 25 entities in both categories) we consider that model obtained couldn't be reliable. What could contribute to positive-negative classification is to manually assign weights to words with polarities 4 and 5. Clustering could be one possible approach for distinguishing the highest sentiment levels, due to the lack of data samples for a classification.

References

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

Jože Bučar. 2017. 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 resources 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.