UNIVERSITY OF TARTU

Institute of Computer Science

Computer Science Curriculum

Olha Shepelenko

Opinion Mining and Sentiment Analysis using Bayesian and Neural Networks Approaches

Master's Thesis (30 ECTS)

Supervisor: Amnir Hadachi, PhD

Supervisor: Artjom Lind, MSc

Tartu – 2017

Opinion Mining and Sentiment Analysis using Bayesian and Neural Networks Approaches

Abstract:

Information technologies have firmly entered our life and it is impossible to imagine our life without gadgets or the Internet. Today, social media is not only a source that broadcasts information to the users, but it allows users to intercommunicate and share their views and experience with each other. Some portion of such data is subjective and contains opinionated information that can be further analyzed to retrieve essential data from it and later use for various purposes for analysis and decision support. In order to use this type of that the first step is to understand it and categorize opinions in the information. Hence, in this dissertation, sentiment analysis techniques are studied in order to retrieve opinions from the tweets. In order to ensure efficient classification, it is important to apply algorithms that perform well on this task. Therefore, the main goal of the thesis is to investigate algorithms that can be applied for the opinion estimation. To that extend, data preprocessing and several experiments are conducted, namely, the classifier is trained and tested on two different datasets with two different classifiers (Naive Bayes and convolutional neural network). In addition, the influence of the training data on the classifier efficiency is discussed.

Keywords:

Sentiment analysis, sentiment, polarity classification, feature selection, Naïve Bayes, convolutional neural network.

CERCS: P170

Arvamuskaeve ja meelsusanalüüs kasutades Bayesi meetodit ja tehisnärvivõrke

Lühikokkuvõte:

Infotehnoloogiad on muutunud suureks osaks meie elust ja praeguseks on raske kujutada ette elu ilma vidinate ja internetita. Sotsiaalmeedia ei ole tänapäeval ainult informatsiooniallikas, vaid lubab kasutajatel ka omavahel suhelda ning jagada üksteisega arvamusi ja kogemusi. Teatud osa sellest infost on subjektiivne ning sisaldab kasutaja seisukohtadega seostuvat informatsiooni. Säärast informatsiooni analüüsides saab sellest eraldada kõige olulisema ning hiljem kasutada saadud informatsiooni analüüsimiseks ja otsuste tegemistes. Esmalt, et informatsiooni sellisel kujul kasutada, on vaja seda mõista ja kategoriseerida. Käesolevas töös õpitakse seisukohtade analüüsimise tehnikaid, et siis säutsudest arvamusi eraldada. Efektiivseks klassifitseerimiseks on oluline rakendada ülesande lahendamiseks algoritme, mis saavad sellega edukalt hakkama. Magistritöö põhieesmärgiks on uurida algoritme, mida saaks kasutada seisukohtade hindamiseks. Teostatakse andmete eeltöötlust ja viiakse läbi mitmeid eksperimente. Klassifitseerijat treenitakse ja testitakse kahe erineva andmekogu peal kasutades kahte erinevat klassifitseerija implementatsiooni, milleks on naiivne Bayes ja konvolutsiooniline närvivõrk. Lisaks arutatakse klassifitseerija efektiivsuse üle ja mis mõju avaldavad sellele andmed, mille peal seda treenitakse.

Võtmesõnad: meelsusanalüüs, seisukoht, polaarsuse klassifikatsioon, omaduste valik, naiivne Bayes, konvolutsiooniline võrk

CERCS: P170

Acknowledgements

I wish to convey my gratitude to Dr. Amnir Hadachi who has been supervising me during the Master's program. I would like to acknowledge his supportive attitude, amazing guidance, perfect time management and endless encouragement and faith. It was extremely helpful and reassuring to hear "You're almost there" and "Stay positive" from him at times when it was really difficult for me to stay calm and positive.

Furthermore, I am extremely grateful to Artjom Lind who was my advisor as well. I wish to thank him for assistance and expertise as well as patience and time spent on the supervision.

Moreover, I wish to express my thankfulness to my husband Sergii for understanding, help and loving support. Finally, I want to thank my friends for constant encouragement upon this two years of my studies.

Table of Contents

Abbreviations and Acronyms

This section clarifies some terms used in the paper.

SA - sentiment analysis

SO - semantic orientation

POS - part-of-speech tagger

PMI - pointwise mutual information

IR - information retrieval

TF - term frequency

IDF – inverse document frequency

SVM - support vector machine

NN - neural network

NB – Naïve Bayes

RNN - recurrent neural network

CNN - convolutional neural network

NLP - natural language processing

1 Introduction

1.1 General view

The difference between people and machines is that people have an ability to articulate personal opinions and the dream behind Artificial intelligence is to make the machine behave like humans. The field of computer linguistics that analyses opinions is called opinion mining or as it is also called sentiment analysis (SA). Opinion mining is the part of natural language processing that deals with analysis opinions about products, services, and even people. Based on [1], sentiment analysis and opinion mining primarily focus on opinions that convey or imply positive or negative sentiment. To perform an analysis of opinions, opinions have to be extracted.

Nowadays retrieval of opinions became easier because individuals share their views about different topics through social networks such as Twitter, Facebook or they leave comments and reviews regarding products on a particular websites. Microblogging is extremely popular way of sharing thoughts and it produces a huge amount of messages every day. Hence, microblogging can be considered as a rich source of opinionated messages that can be collected and further utilized for extracting sentiments. Analysis of opinions plays an important role in all science areas (politics, economics, and social life). For example, in marketing, if the seller knows about the customer's satisfaction of particular product he/she may estimate demand on the product. The same for politicians, they will know whether people support them or not.

Sentiment classification task is not new research area. However, the main focus of research was on the analysis of big documents (reviews), but not on microblogs that are sought-after today. Twitter is one example of microblogging platform. The tweet is a short message (maximum 140 characters) that can contain opinion or just express some facts. Classification of tweets is a difficult task because tweets can contain irony, misspellings, emoticons, slang, abbreviations and it may contain only a few words. Let us consider the following tweet example: “Nice restaurant, yummyyyyyy meal, warm atmosphere, although the klutzy waitress spill vine on my dress :[#screwup “. Tweet contains elongated word (yummyyyyyy), misspelled (vine instead of wine), emoticon (“ :[“), slang (#screwup, klutzy), also it holds both positive and negative opinions. All these factors complicate the process of classification.

Various techniques exist that can be used for sentiment analysis task. The main approaches are machine learning [1], [2], [21], [24], [32] and lexicon-based [1], [2], [11], [12], [18]. Machine learning approach uses dataset for training classifier which will be further applied for defining sentiment of a particular text. The lexicon-based method uses the semantic orientation (SO) of words or phrases to define whether a text is positive or negative.

In this thesis, Twitter platform will be used as a source of opinions. Sentiment analysis and mining approaches will be utilized for sentiment extraction. This dissertation focuses on the analysis of different techniques for composing features set that will be used for training and testing network. Moreover, main interest of our investigation is to find efficient algorithms that can be applied for classification purposes on tweets. In this dissertation, machine learning algorithms will be applied for sentiment classification. Important to notice that supervised learning is the most applied technique for sentiment classification. To be precise, the main focus in this thesis is on Naïve Bayes algorithm and Convolutional Neural Networks as classification methods.

1.2 Objectives and Limitation

The aim of this thesis work is to investigate and discover efficient algorithms that can be used for sentiment analysis as well as to provide improvements for existing solutions. To that extend, following steps should be done:

- Investigate feature selection methods for text classification.
- Investigate machine learning algorithms that can be applied for classification problem.
- Analyze accuracy of these algorithms with respect to different datasets
- Analyze computational time of the algorithms and computational resources that particular algorithm requires.
- Compare results of applied techniques.
- Based on obtained result, propose a set of improvements.

While working on this thesis work, limitations were found. First of all, the neural network requires huge dataset to be fed to the system to train it efficiently in order to gain good results. In addition, it is difficult to find large labeled datasets on the Internet. Data can be extracted from the Twitter regarding a specific domain, but labeling these data will require a lot of time. Another drawback of the available Twitter dataset is that it contain noisy data that have to be removed.

Next limitation is the computational resources. Working with large dataset require powerful machines to process these data. It is especially critical when training and testing neural network.

1.3 Contributions

The main contribution of this research is an investigation of classification algorithms for extracting opinions from tweets and movie reviews. For this purpose, two methods are studied. First is Naïve Bayes algorithm that uses a bag-of-words representation for training classifier. Second is a convolutional neural network that converts words into word embeddings and then passes these embeddings through the layers to extract the polarity of tweets. Therefore, the aim of the thesis is to perform experiments and investigate the performance of two different algorithms detecting positive and negative tweets/reviews. Furthermore, algorithm which gives better results has to be defined. In addition, it is important to study how algorithms accuracy can be affected by data preprocessing, feature selection and data selection.

To apply machine learning algorithms several steps should be performed:

1. Data collection. Tweets to be analyzed have to be retrieved from Twitter as well as the dataset for training purpose has to be obtained.
2. Preprocessing data. Tweets have to be pre-processed in order to remove the usernames, URLs, punctuation that do not contain any useful information. Moreover, words have to be lowercased.
3. Training process. Data that was extracted as training set is given to the classifier for learning.
4. Data classification. When training stage is complete the classifier can be used for analyzing polarity of tweets or reviews. At first, the classifier is fed with the testing dataset to check the accuracy of the algorithm then real data can be given to the classifier to extract sentiments from tweets.

After machine learning algorithms are applied results are analyzed. Namely, accuracy of algorithms and their performance time are analyzed. Depending on results recommendations for improvement classification process is given.

1.4 Road Map

The rest of the thesis has four more chapters and organized as follows.

Chapter 2: Describes different methods that can be applied for sentiment classification task. The literature review includes discussion on machine learning and lexicon-based approaches. Definition and challenges of sentiment analysis are introduced and commonly used algorithms for defining polarity of tweets are defined.

Chapter 3: Gives information about the data used for training and testing the classifiers, also importance of data preprocessing is explained. Moreover, feature selection mechanism is presented. The main part of this chapter is an overview of used classification algorithms and their implementation details.

Chapter 4: Presents experiments and results achieved in this research work. Namely, testing of applied algorithms. Based on the obtained results analysis is performed in order to define which of the algorithms performs better for tweets classification.

Chapter 5: Conclusion and future research perspectives are presented in this chapter.

2 Literature review

This chapter provides an overview of the different approaches that can be applied for sentiment analysis as well as brief explanations of algorithms used by researchers.

2.1 Introduction

Sentiment analysis is not a new task, it has been studied since 90s. However, in 2000s SA attracted the interest of scientists due to its significance in different scientific areas, also SA had a many unstudied research questions [1]. Moreover, the wide availability of opinionated data pushed research in this area on a new stage. Since then SA became rapidly developing area.

According to Bing Liu in [1]:

> "*sentiment analysis, also called opinion mining, is the field of study that analyzes people's opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes*."

In other words, sentiment analysis deals with processing of opinionated text in order to extract and categorize opinions from certain document. The polarity of sentiment usually expressed in terms of positive or negative opinion (binary classification [3], [4]). However, it can be multi-class classification [5], [6], [7], hence sentiment may have a neutral label or even broadened variation of labels like very positive, positive, neutral, negative, very negative, also labels can be associated with emotions like sad, anger, fearful, happy, etc.

Sentiment analysis is a developing area that arouses the interest of humans and especially organizations because SA can be used for decision making process. Individuals are no longer limited to ask opinions from friends about particular product or service, they can freely find such information on the Internet. Furthermore, organizations may save time and money by avoiding of conducting surveys instead they can concentrate on processing opinions that can be obtained from the Web freely. Nevertheless, it is important to notice that sources that contain opinionated data

are noisy sometimes, so it is important to extract the essential meaning from that information to use it further. SA uses different techniques and approaches for handling this challenging task [1].

Sentiment analysis can be carried out at the following levels:

- Document level [8]. At this level the main task is to define opinion of the whole document (opinion should be expressed about one topic).
- Sentence level [4]. Here every sentence is considered as a short document which can be subjective or objective. Subjective (opinionated) sentence expresses sentiment.
- Aspect level (feature level) [9]. Allows to extract opinions towards aspects of entities.

Sentiment analysis classification techniques mainly divided into machine learning and lexicon-based approaches [2] (see Figure 1). More detailed explanations of these techniques will be given in the following subsection.

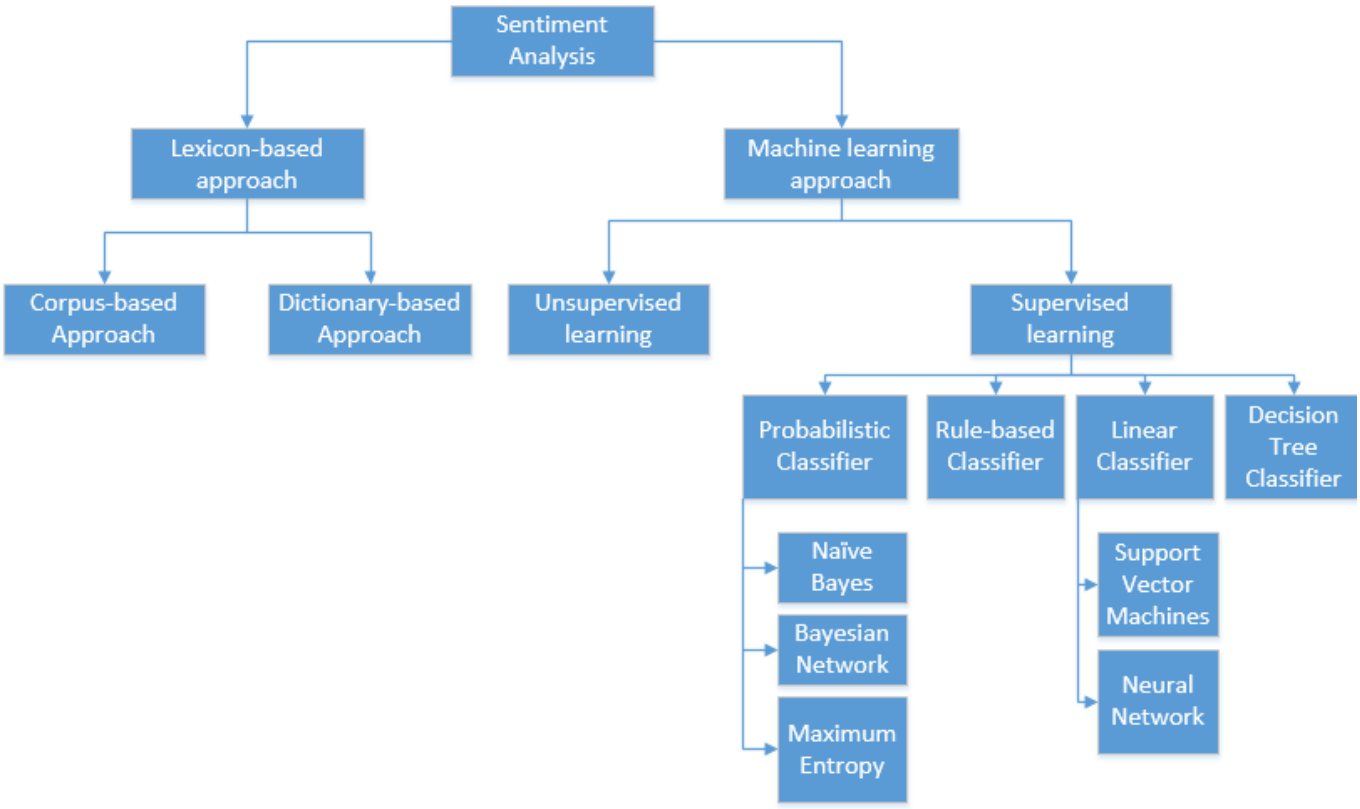

Figure 1. Sentiment analysis approaches (inspired from [2]).

2.2 Literature

This section focuses on describing lexicon-based and machine learning approaches.

2.2.1 Lexicon-based approach

The first technique that can be used for SA is the lexicon-based method. It uses a lexicon that consists of terms with respective sentiment scores to each term. The term can be associated with a single word, phrase or idiom [10]. The sentiment is defined based on the presence or absence of terms in the lexicon. The lexicon-based approach includes corpus-based approach and dictionary-based approach that are discussed further.

2.2.1.1 Dictionary-based approach

The main idea behind the dictionary-based approach is to use lexical databases with opinion words to extract sentiment from the document. Based on [1], [11], a set of seed sentiment words (e.g. good, bad) with their polarities is collected by hand. At the beginning, this initial set does not have to be large, 30 opinion words is enough [12]. Next step is to use the polar words to enrich a set by looking up for respective synonyms and antonyms in a lexical database. Examples of such databases are WordNet [13], HowNet [14], SentiWordNet [15], SenticNet [15], MPQA [15], etc. The look-up procedure is iterative. At each iteration the algorithm takes updated set of words (expanded set) and does search again until there will be no new words to include. In the end, a set of sentiment words can be reviewed with a purpose of deleting errors.

Hu and Liu [12] have focused their research on the classification of customer reviews, namely they extracted product features that contain sentiments, then classified sentences based on that features and as a result, the summary of the product reviews was composed. For example, if a review was about a camera, authors retrieved such features as picture quality and size of the camera, and using these features, the classification was made on positive and negative camera reviews. In order to assign a positive or negative tag for a sentence, first, researchers retrieved polar words from each review. In this case, adjectives were used. The prediction was based on the polarity of an adjective which had the same the polarity as its synonyms and opposite to polarity of its antonyms. Polar words were utilized for searching their synonyms and antonyms with known

orientation in WordNet. Therefore, the orientation of polar words that appear in the review was identified. The method that was described in [12] showed good results, average accuracy constituted 84%. Hence, current method can be effective for prediction of adjective semantic orientations and sentence polarity.

Kim and Hovy [16] investigated the sentiment of the text and its holder regarding a given topic. Authors of the research paper [16] have applied several classifiers. The first classifier was applied to each word in the sentence to get its polarity. The second classifier defined the polarity of the entire sentence expressed by opinion holder. In addition, the authors introduced the use of small initial list of seed words in a similar way as in [12] (adjective and verbs). This latter was extended by looking up for corresponding synonyms and antonyms in WordNet. Authors mentioned that some synonyms/antonyms had neutral or even opposite orientation that makes them inappropriate to use. Moreover, the researchers emphasized the necessity of defining the strength of positiveness and negativeness of the words that would allow to eliminate ambiguous words. Kim and Hovy identified the four different regions in the sentence that are close to opinion holder and can contain sentiment. For determining the sentence orientation, authors developed three models. First model was based on the assumption that "negatives cancel one another out" [16]. Second and third models were the harmonic and geometric mean of the sentiment strengths in the particular region respectively. After conducting experiments it was concluded that the best results retrieved by using first model and region that starts from opinion holder to the end of the sentence.

In the paper [17] authors developed a method that was using three various dictionaries (traditionally only one is used) to obtain synonyms and antonyms based on seed words. Afterwards expanded lexicon was used for tweets classification. Authors said that their proposed technique made possible to classify tweets that traditional dictionary-based method was not capable. Nevertheless, suggested approach has several drawbacks. The main problem is a collection of synonyms and antonyms require a lot of time. Also, usually dictionaries contain formal words, but tweets are full of informal lexis.

Generally, the main drawback of dictionary-based approach is the inability to detect sentiment words with domain and context specific polarity orientations [2].

2.2.1.2 Corpus-based approach

In [1] Bing Liu indicates that corpus-based approach can be applied in two cases. First case is an identification of opinion words and their polarities in the domain corpus using a given set of opinion words. The second case is for building a new lexicon within the particular domain from another lexicon using a domain corpus. The findings suggest that even if opinion words are domain-dependent it can happen that the same word will have opposite orientation depending on context.

The research conducted by Hazivassiloglou and McKeown [18] is prominent in the literature about corpus-based technique. Authors proposed a method that extracts semantic orientation of conjoined adjectives from the corpus. The technique is based on the usage of textual corpora and seed opinion words (adjectives). Special linguistic rules are applied to the corpora in order to discover opinion words with corresponding polarities. Authors assume that adjectives have the same polarity if they are joined by the conjunction "and". However, the conjunction "but" is used for linking adjectives with opposite polarities. Additionally such conjunctions as "or", "either-or", "neither-nor" are used. Sometimes these rules do not applicable. Therefore, authors also predict the polarities of the conjoined adjectives to check whether the polarities are the same or not, for this purpose log-linear regression model is used. After prediction stage, the graph is obtained that provides links between adjectives. Then clustering is carried out on the graph to divide adjectives into positive and negative subsets. To conclude, Hazivassiloglou and McKeown were able to achieve 90% precision.

As mentioned above, the same sentiment word can have different semantic orientation depending on the context. Ding et al. [19] proposed a method for finding the orientation of sentiment conveyed by reviewers. Authors have emphasized that some adjectives (mostly quantifiers, like long, short, etc.) are context-dependent and can change their polarities. Researchers consider sentiment words with their aspects in the sentence in order to identify the polarity of the product feature. Ding et al. use words, phrases, and idioms as an opinion lexicon. List of adjectives and adverbs is taken from [12] and extended by authors to include verbs and nouns. Moreover, they annotated around 1000 idioms that contain clearly expressed sentiment. After the lexicon is ready, they define the polarity score for each feature in the review sentence. To get the score for the whole sentence they sum up all the scores using proposed score function

that gives better results than simple summation used in [12]. Additionally, authors applied several linguistic rules for handling negations and sentences that contain the conjunction “but”. Furthermore, the paper introduces a holistic approach to solve the problem of the identifying polarity of context dependent sentiment words. For this purpose, three consistency techniques about connectivity are suggested [19]: intra-sentence conjunction technique, pseudo intra-sentence conjunction technique, and inter-sentence conjunction technique. To sum up, the authors report that the proposed approach is effective and gives better results than previously proposed methods.

The corpus-based method alone is less effective that dictionary-based method due to a limitation of words that are in the corpus. However, usage of this approach can help to construct domain and context specific lexicon.

Overall, the performance of lexicon-based methods in terms of time complexity and accuracy heavily depend on the number of words in the dictionary, namely, performance decreases significantly with the exponential growth of the dictionary size [20].

2.2.2 Machine learning approach

The second technique that can be used for SA is machine learning that includes unsupervised and supervised machine learning methods that are explained below.

2.2.2.1 Unsupervised machine learning methods

Unsupervised learning approach uses unlabeled datasets in order to discover the structure and find the similar patterns from the input data. Unsupervised method is usually used when a collection of reliable annotated dataset is difficult, but collecting of unlabeled data is easier. It does not cause any difficulties when new domain-dependent data have to be retrieved.

Turney [21] uses unsupervised machine learning approach for the reviews classification. Reviews are classified into recommended (thumbs up) and not recommended (thumbs down). The author retrieves phrases that consist of two words based on tags patterns. The patterns are designed in such a way that they have to capture sentiment phrases. Each phrase is a combination of adjective/adverb and verb/noun (overall, 5 patterns are proposed). Part-of-speech tagger (POS) is employed to the document in order to decide which phrases have to be retrieved. Note that a phrase is extracted if two words fall under one of the proposed patterns. Next step is a calculation of

semantic orientation of retrieved phrases from the review. The author applies Pointwise Mutual Information and Information Retrieval algorithm (PMI-IR) to find semantic orientation. PMI measures semantic similarity between two terms. A phrase that conforms to the patterns is taken as a first term and a reference word is taken as a second term. “Excellent” and “poor” words are considered as the reference words because it is natural to grade a review as poor when it gets one star and “excellent” if review receives five stars. The semantic orientation of a phrase is defined as a difference between PMI (phrase, “excellent”) and PMI (phrase, “poor”). Semantic orientation is positive if a phrase has a stronger association with “excellent” reference word and negative if an association is stronger with “poor”. To calculate PMI the co-occurrence probabilities of respective terms have to be defined. For this purpose number of hits is estimated. The number of documents (hits) that contain a first term and a second term separately as well as two terms together are returned by AltaVista search engine when searching these terms in it. The last step is to find the sentiment for the whole review (recommended or not recommended). If the average semantic orientation is positive the review is marked as recommended and not recommended otherwise. As reported, the average accuracy constituted 74%.

Rothfels and Tibshirani [22] applied an unsupervised method for sentiment classification of movie reviews. Authors adapted the method that was proposed by Zagibalov and Carroll [23] for classification of Chinese text. The idea of the method is to use positive seed words that can be retrieved from the document. Such sentiment words (adverbs) are preceding negations or can occur without negation (most common case). Having an initial seed set Zagibalov and Carroll [23] have enriched the list of positive seed words applying iterative classification. Inspired by Zagibalov and Carroll’s approach the authors of paper [22] composed an initial seed set. The text of the document to be classified was divided into zones, each zone corresponds to the piece of text located between punctuation characters. Then classification of each zone was performed. The sentiment of the whole text is defined by the predominance of positive or negative zones in the document. Namely, if positive zones occur more frequently than negative zones then the document is recognized as positive and negative otherwise. Rothfels and Tibshirani also expanded the list of seed words. They tried to use bigrams, trigrams, and 4-grams as seed words. However, first two did not preserve the content of phrase opposed to 4-grams. This latter still gave unsatisfactory results. Authors [22] made the second attempt. They have used semantically meaningful words (adjectives) as a seed set. Nevertheless, improvement of accuracy was not achieved as was

expected. Researchers also tried to change the scoring method to k-means clustering, but obtained results did not show any essential enhancement. In their last attempt, they adapted approach proposed in [21], which estimates the semantic orientation of a phrase. Rothfels and Tibshirani manually composed 2 seed sets (positive and negative). After that, the semantic orientation between each word in the text and a reference seeds was estimated. The final list of sentiment seed words contains only words with the high semantic score. The last step of the adapted algorithm is an iterative classification using SO of each word as it initial sentiment score. This time accuracy increased almost by 15%.

2.2.2.2 Supervised machine learning methods

Supervised machine learning methods assume the presence of labeled training data that are used for the learning process. The latter estimates the output from the input dataset, we refer to the case when the classifier defines the label the object belongs to. As training data set, labeled documents have to be used. Usually, bag-of-words model [24] is employed to represent a document as a feature vector $d = (w_1, w_2, \dots, w_i, \dots, w_N)$, where N is set of all the unique terms in the training dataset and w_i is weight of the i-th term. To convert training dataset to a feature vector, vocabulary with N unique words has to be created from the training data. Further, any of feature models can be used for constructing a feature vector itself. Examples of feature models are [24]:

- Binary feature model. w_i is assigned to 1 if the term is present in the document, otherwise 0.
- Term frequency (TF) defines the number of times a term occurred in the document.
- TF-IDF (IDF – inverse document frequency). IDF measures the importance of a term (TF considers that all the terms are equally important).
- Information gain (IG) estimates a prevalence of the feature in particular class compared to other classes. IG allows to use the terms that are highly informative.
- Chi-square test will be considered in Chapter 3.

After the dataset is represented as a vector, it can be used by the classifier for learning and estimating labels. Different kind of methods can be used for training the classifier. Let's discuss some of them.

The most common and simple method that is used for text classification is Naïve Bayes [24], [25], [26], [27]. The model is based on Bayes' theorem with the assumption that features are independent. Naïve Bayes classifier defines the probability of the document belonging to a particular class. The advantages of the Bayes classifier are: simplicity of the implementation, learning process is quite fast, it also gives quite good results [4], [26], [27]. However, "naive" assumption may cause a problem because in the real world features are dependent.

According to [25] "the idea behind Maximum Entropy models is that one should prefer the most uniform models that satisfy a given constraint". The probability of the document belonging to a particular class [25], [26] estimated as:

$$P(c|d,\lambda) = \frac{\exp[\sum_i \lambda_i f_i(c,d)]}{\sum_{c'} \exp[\sum_i \lambda_i f_i(c,d)]}$$

Where, c is the class, d is the document to be classified, λ is a weight of the *i*-th classification indicator f_i. Maximum Entropy classifier does not assume independence of features. Thus such classifier theoretically may outperform Naïve Bayes. However, Maximum Entropy algorithm is more difficult to implement and learning process is slower.

Another approach for classification is rule-based. The idea behind the method is to apply a set of rules that were generated by experts based on the analysis of the domain-specific area. This method can show good results when using a wide range of rules. However, the creation of such rules is time-consuming. Rule-based approach was used by Chikersal et al. [28]. They proposed the rules that depend on the occurrence of emoticons and sentiment words in tweets. In addition authors [28] applied Support Vector Machine (SVM) classifier (for SVM classifier emoticons were removed from the training dataset). They used a linear kernel and L1-regularization in all experiments. Authors employed varied features like the word *n*-grams, POS-tags, character *n*-grams as well as different lexicons: Bing Liu lexicon, Sentiment140 lexicon, SentiWordNet, etc. The idea of their approach is to combine two methods in order to improve precision and recall. Each tweet that was labeled as neutral by SVM classifier further analyzed by rule-based classifier for receiving the final label. Summing up the results, it can be concluded that rule-based approach can improve the prediction made with SVM classifier.

SVM classifier was also used in [25], [26], [27]. The method assumes a division of space into subspaces that correspond to particular classes. In terms of binary classification, idea of the

training stage is to discover a hyperplane that best separates a dataset into two classes with the maximum margin. The margin is the distance from the hyperplane to the closest data point from the set defined by the hyperplane. These data points that are close to the hyperplane are called support vectors. This latter are critical elements, because removal of them would change the position of the separator [29]. To conclude, SVM sometimes can outperform such algorithms as Naïve Bayes, Maximum Entropy [27]. However, SVM is not suited to large datasets due to SVM's time complexity.

Another solution for text classification is the usage of Neural Networks (NN). Artificial neural network follows principles of the biological neural network. It is assumed that neural network can solve the issues in the same way as they can be solved by humans. The NN is a collection of interconnected neurons. Usually, NN has multiple layers. The neural network is able to learn through adjusting the weights of the neurons. Consider the following types of the neural network that was used for text classification:

- Convolutional neural network (CNN). CNN is organized by layers interleaving. Such network contains convolution, subsampling and fully-connected layers that can alternate in random order. Severyn and Moschitti [30] were working on Twitter sentiment analysis with deep CNN. They proposed a one layer network that includes a convolutional layer that is passed through the non-linear activation function (ReLU) followed by max-pooling layer and further passed to soft-max classification layer. Neural language model (that was proposed by T.Mikolov et al. [31]) was used for initializing word embedding out of an initial dataset of tweets. Then word embeddings were refined using CNN on the distant supervised corpus. Authors claim that proposed system performs well.

  Moreover, Kim [32] was using CNN for sentence classification. He classified sentence into positive/negative as well into fine-grained classes, also he defined whether a sentence is subjective or objective and classified a question into 6 question categories. For this purpose different test data sets were used. As in previous research [30], one layer CNN is employed and word embeddings gained from an unsupervised neural language model [31]. CNN includes convolutional, max-over-time-pooling and fully connected layers. It was reported that model

showed good results and “pre-trained vectors are ‘universal’ feature extractors that can be utilized for various classification tasks” [31].

Similar architecture that showed high performance was also described in [33].

- Recurrent neural network (RNN). RNN is a network with feedback connections that allows to keep information about the previous moment of time. In this type of the network, the output that was computed in the previous step is used for computing the next. Then the output is compared with test data and an error rate is estimated based on which weights are adjusted that makes the learning process more accurate. RNN is useful with sequential information for predicting next word in the sentence [34]. This property allows to get a better understanding of the sentence by capturing the context of every word based on previous ones. Pengfei Liu et al. [35] have employed RNN for text classification with multi-task learning. As the tasks, they chose the following: the 5 class classification, binary classification, the classification the sentence into subjective or objective (sentence-level) and binary classification on document-level. In the article [35] authors presented 3 architectures of sharing information to model text sequence. The first architecture utilizes one shared layer for all tasks. The second architecture utilizes different layers for different tasks. The last model assumes the assignment a certain task to a certain level, but also have a shared layer for all the tasks. After experiments were conducted, authors compared obtained results and concluded that on some task they achieved better results opposed to the state-of-the-art baselines.

Decision tree is another way to perform classification. Decision tree [36] is a classifier that is presented as hierarchical decomposition of data space. The tree structure contains 2 types of nodes: leaf node (contains the value of the target attribute, i.e. positive or negative label in binary classification task) and decision node (contains a condition on one of the attributes for space division). The partitioning of the data space is done recursively.

2.4 Conclusion

This chapter provides brief explanations of algorithms that can be applied for the classification task. Moreover, approaches that were used by researchers for text classification are examined. More specifically, a variety of lexicon-based and machine learning methods are discussed. It has been shown that supervised machine learning approaches opposed to unsupervised methods show good results when dealing with sentiment classification [27]. Naïve Bayes classifier performs well on text classification despite its simplicity. Furthermore, Convolutional neural network is also effective for natural language processing (NLP), it allows to significantly reduce the number of learning parameters and obtain a high quality of classification. Based on these factors, supervised methods such as Naïve Bayes and CNN are further employed for sentiment analysis on tweets and movie reviews.

3 Methodology and contribution

3.1 Introduction

Nowadays, we cannot imagine our life without accessing the World Wide Web, everyone uses Internet for different purposes, i.e. for searching some information or posting something. Information can be easily published by users in the blogs, forums, social networks, feedbacks can be left on the particular web pages. There are a lot of sites that provide business and product reviews. For example, Amazon is an e-shop where customers can publish their feedback about products as well as look up for reviews to make a decision for purchasing a product. Another interesting and useful source of opinions is TripAdvisor. TripAdvisor is a website that provides dozens of opinionated information about hotels, restaurants, flights, places where to go, which is very helpful for travelers. Twitter is another way of sharing views. Information from such sources is used not only by customers, but it is also vital for different organizations. The wide availability of opinionated data caused the necessity of creation of an automated system for searching and classifying opinions.

Text classification task is not a new area of study; however, mostly research conducted was performed on the short texts like product [12], [19], [21] and movie reviews [21], [22]. Concerning Twitter, the messages differ from reviews by their length (140 characters) and special symbols like @, # or RT that they include. Moreover, the way of chatting is informal that leads to the usage of slang and idioms, the misspelling is also common for such type of text. Certainly, tweet classification was also studied [28], [30], [32], but less research on tweets was carried out opposed to reviews.

3.2 Problem statement

Focus of this dissertation work is to conduct sentiment analysis on Twitter messages and movie reviews by identifying positive and negative ones. This latter will be done by applying two different approaches. To be more precise, the interest is in investigating the approaches for polarity classification and choosing the most efficient one. Notice, based on the literature review as

discussed in chapter two, we decided to investigated two approaches in details in this dissertation: Naïve Bayes and Convolutional neural network, since the trend in the scientific community is focused on this type of methods. Classification is performed on tweets, where each tweet is labeled as positive or negative according to the opinion expressed in it.

Before building the classifier, training data have to be collected and preprocessed in order to discard irrelevant information from the training set. Moreover, preprocessing should be performed to reduce the size of training dataset, which in turn may lead to speeding up the training process. The next important step that has to be done until training the model is feature selection. Feature selection allows to build a set of unique terms (features) across the corpus by excluding ambiguous terms. After the classification model is created and tested, parameters of accuracy, precision and recall as well as computation time have to be estimated. Furthermore, comparison of the applied algorithms has to be provided according to the classification results.

3.3 Methodology

This subsection introduces the main steps that have to be performed for carrying out the sentiment classification, namely preprocessing and feature extraction. Moreover, two algorithms that are used for classification described in details.

3.3.1 Data and preprocessing

In this dissertation work, supervised methods are employed. These methods require labeled training dataset. Two datasets are used for training classifiers. The first dataset is a dataset v1.0 introduced in Pang/Lee ACL 2005 [37] that represents movie reviews. Dataset includes 10662 automatically labeled reviews, half of them are positive reviews and another are negative. Dataset does not have split on training and testing data. Therefore, 90% of data is taken as training data for creating supervised learning model based on Naïve Bayes and neural networks, 10% is taken as test set for estimation the accuracy of the classifiers. The dataset of movie reviews is considered because such kind of reviews comprise a broad range of emotions and capture many adjectives suitable for sentiment classification. The second dataset is a dataset that contains automatically annotated tweets [38]. This dataset was collected by Go et al. [3], their approach based on usage

of emoticons (":)", ":-)", ":)", ":D", "=)" mapped to positive emoticons and ":(", ":-(", ": (" mapped to negative). The total amount of tweets in the second dataset constitutes 1.6 million tweets, dataset evenly contains positive and negative tweets. The testing data includes 359 manually annotated tweets, which are labeled as positive and negative. The statistics of the datasets are presented in Table 1.

Table 1. The statistics of the datasets.

Dataset	Type	Positive	Negative	Total number of tweets
Movie reviews [37]	Train	4 797	4 797	9 594
	Test	534	534	1 068
Tweets [38]	Train	800 000	800 000	1 600 000
	Test	182	177	359

After training data is extracted, next step is to preprocess it in order to exclude irrelevant data from the dataset. Preprocessing is crucial in terms of computation time and classifier performance because noisy data can slower the learning process and decrease the efficiency of the system in general. Preprocessing includes the following:

- Removal of URLs. Frequently tweets contain web links to share some additional information. The content of the links is not analyzed, hence address of the link itself does not provide any useful information and its elimination can reduce the feature size, which is why URL is removed from the tweet.
- Removal of usernames. Another user can be mentioned by post creator in the tweet by using "@" symbol followed by username, i.e. @Superman. Due to this feature does not provide any relevant information it was also excluded from the tweet.
- Removal of hashtags. The hashtag is depicted using "#" symbol and used before a word that represents a topic name. Topics are not the task to be classified, hence they are omitted.
- Removal retweets and duplicates. The retweet is a tweet that is written by one user and then copied and posted by another user. Retweet contains "RT" abbreviation. Repeated tweets and retweets are removed in order to exclude putting extra weight on a specific tweet.

- Compression of elongated words. It is usual to elongate words to express sentiment, i.e “I am so lovelyyyyyyyyyy”, that conveys positive sentiment and repeated letters emphasize the strength of positiveness. It depends on the user how many letters to repeat. In order to avoid hundreds of representations of the same word, it was decided to compress the repetition of the same letter. If the letter occurs more than three times it normalized to the sequence of three letters, i.e “lovelyyyyyyyyyy” is converted to “lovelyyy”.
- Removal of stop words. Stop words are extremely frequent words that considered as valueless for taking them as features, i.e. “the”, “for”, “her”, “a”, etc. Stop words are discarded from the tweets.
- Lower casing is necessary in order to ensure that the term (in our case word) mapped to respective feature, i.e. “Happy” and “HaPpY” should be mapped to “happy”. This step guarantees consistency within feature set.

3.3.2 Feature extraction

After preprocessing is completed, features have to be extracted and further used for training the classifiers. In the first experiment, the unigrams were selected as features for feeding the Naïve Bayes classifier. Sentence (movie review/ tweet) is split into words (unigrams) and represented as a set of words. Using unigrams end up in a large feature set that has to be reduced to eliminate uninformative features.

Chi-square feature selection algorithms [2], [29] was investigated for the Naïve Bayes model. Chi-square is a statistical test that measures the independence between the class label and the feature itself. It estimates the importance of the terms by calculating their scores, in other words, it measures the correlation between terms and their classes. Chi-square can be calculated using the following formula [29]:

$$\chi^2(\mathrm{D}, t, c) = \sum_{e_t \in \{0,1\}} \sum_{e_c \in \{0,1\}} \frac{(N_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}}$$

where $\mathbb{D}$ is training set,

N is observed frequency in $\mathbb{D}$, E is expected frequency,

$e_t = 1$ if document contains the term t,

$e_t = 0$ if document does not contains the term t,

$e_c = 1$ if document is in the class c,

$e_c = 0$ if document is not in the class c.

Let's rewrite above formula as:

$$\chi^2(D, t, c) = \frac{(N_{00} + N_{01} + N_{10} + N_{11}) * (N_{11}N_{00} - N_{10}N_{01})^2}{(N_{11} + N_{01}) * (N_{11} + N_{10}) * (N_{10} + N_{00}) * (N_{01} + N_{00})}$$

where N is the total number of training instances, $N = N_{00} + N_{01} + N_{10} + N_{11}$,

N_{00} is the number of sentences that does not contain feature ($e_t = 0$) and are not in the class ($e_c = 0$),

N_{11} is the number of co-occurrence of the feature and class,

N_{10} is the number of sentences that contain the feature and are not in class,

N_{01} is the number of sentences in class but does not contain feature.

The large value of chi-square implies that two event are not independent (namely, class label and feature are dependent) that means the null hypothesis of independence should be rejected. If events are dependent then the feature is picked. Hence, knowing the chi-square score allows choosing the most informative feature for learning classifier.

The second experiment was conducted using a convolutional neural network. CNN uses filters (kernels) that play the role of feature detectors. Using initial dataset a vocabulary *V* has to be formed, where each word is indexed (index is an integer that lies in the range from 0 to the vocabulary size). In this dissertation work two different datasets are used, the size of the movie reviews dictionary constitutes 18758 words and size of the tweets dictionary constitutes 274562 words. After building the vocabulary, the first layers of the CNN represented as the low-dimensional vectors. Namely, each movie review or tweet (sentence implies to the movie review or the tweet) is handled as a sequence of words $s = [s_1, s_2, \dots, s_s]$, where each word is mapped into the index corresponding to this word in the vocabulary *V*. The sentences of varied length normalized by padding them to the maximum length of the sentence. Overall, each sentence is converted to the vector representation and the whole input text is represented as a matrix (see Figure 2). To feed the latter to the convolutional layer, it has to be further converted to the

embeddings that are stored in a lookup table (also see figure 1). In this dissertation word embeddings initialized randomly. Word embeddings are used as an input to the convolutional layer.

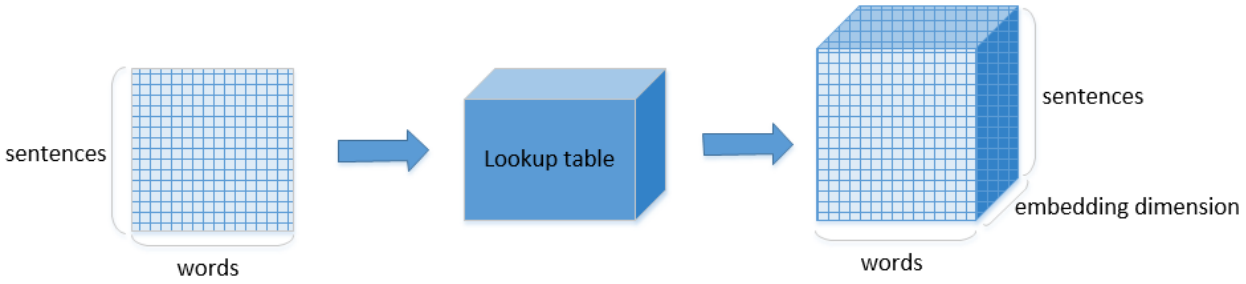

Figure 2. Transformation of input text.

To select informative features from the initial dataset and move to higher level perspective, convolution and pooling operations have to be employed. These operations will be introduced in Subsection 3.3.3.2.

3.3.3 Classification algorithms

This subsection provides detailed explanation of two algorithms that are exploited in the dissertation.

3.3.3.1 Naïve Bayes approach

Naïve Bayes classifier has shown its efficiency and simplicity in applying it for sentiment classification [4], [26], [27]. It is a probabilistic approach integrating the Bayes' algorithm [43] that allows to compute probability of features belonging to a label:

$$P(label|features) = \frac{P(label) * P\ (features|label)}{P\ (features)}$$

where $P(label|features)$ is the posterior probability of features belonging to a label (positive or negative),

$P(label)$ is the prior probability of a given label,

$P\ (features|label)$ is the conditional probability that the particular feature in features appears given label,

$P\ (features)$ is the prior probability of the feature in features.

Let's make 'naïve' assumption that the features are independent of each other. That gives the following:

$$P\ (features|label)\ \approx P(f_1|label) *\ P(f_2|label) * \ldots *\ P(f_n|label) = \prod_{i=1}^{n} P(f_i|label)$$

$$P(label|features) = \frac{P(label) * \prod_{i=1}^{n} P(f_i|label)}{P\ (features)}$$

where f_i is an individual feature.

Although Naïve Bayes model assumes that features are generated independently of their positions, it still gives good result in real tasks.

The main goal of the classification is to define the label the feature belongs to. Therefore, we do not interested in finding the probability itself, however, the most probable label has to be defined. Naïve Bayes classifier uses the maximum a posteriori (MAP) estimation to define the most probable label $label_{map}$ [29]:

$$label_{map} = arg \max_{label \in L} \left[\frac{\hat{P}(label) * \prod_{i=1}^{n} \hat{P}(f_i|label)}{\hat{P}(features)} \right]$$

Denominator can be omitted due to it is the same for positive and negative labels. Hence expression for computing $label_{map}$ can be rewritten as:

$$label_{map} = arg \max_{label \in L} \left[\hat{P}(label) * \prod_{i=1}^{n} \hat{P}(f_i|label) \right]$$

P marked as $\hat{P}$ because true values of the corresponding parameters will be estimated from the training dataset [29].

Many conditional probabilities are multiplied in the equation above. Based on [29] the latter may lead to a floating point underflow that can be avoided if employ the logarithm property, namely, $\log(xy) = \log x + \log y$. Now, multiplication of probabilities is represented as the sum of logarithms. Because of logarithm function is monotonic, it can be applied to both parts of the equation without changing the parameters where maximum is achieved, but only numeric value will be changed (that does not cause any issue). Therefore, the label equation will be defined as follows:

$$label_{map} = arg \max_{label \in L} \left[\log \hat{P}(label) + \sum_{i=1}^{n} \hat{P}(f_i|label) \right]$$

As was mentioned, estimation of $P(label)$ and $P(f_i|label)$ is performed on the training dataset. The prior probability $\hat{P}(label)$ can be defined as:

$$\hat{P}(label) = \frac{N_{label}}{N}$$

where N_{label} is the number of features that refers to the respective label, and N is the total number of features in the training dataset.

Conditional probability $\hat{P}(f_i|label)$ can be computed as:

$$\hat{P}(f_i|label) = \frac{F_{i\ label}}{\sum_{i' \in V} F_{i'\ label}}$$

where $F_{i\ label}$ is the number of times the i-th feature occurs in the training dataset with respective label, including repetitions of the features, and V is the dictionary of all unique features considering the label.

Important to notice that on the classification stage the situation can happen when classifier may encounter a new feature that has never occurred in training samples, hence it is unknown for the classifier. In this case classifier will skip this feature. Thereby, final formula for defining the "best" label using Naïve Bayes classifier can be written as:

$$label_{map} = arg \max_{label \in L} \left[\log \frac{N_{label}}{N} + \sum_{i=1}^{n} \log \frac{F_{i\,label}}{\sum_{i' \in V} F_{i'label}} \right]$$

In this dissertation work Multinomial Naïve Bayes classifier is applied (features shows how many times each word occurs in the given dataset). The Naïve Bayes algorithm is represented in Figure 3 (inspired from [29]).

Algorithm 1 Naive Bayes

procedure TRAININGNB(L, F)
  $V \leftarrow ExtractVocabulary$(F)
  $N \leftarrow CountAllFeatures$(F)
  for each $label \in L$ **do**
    $N_{label} \leftarrow CountFeaturesBelongingToTheLabel$(F)
    $priorProb[label] \leftarrow N_{label}/N$
    $text_{label} \leftarrow CombainTextOfAllFeaturestBelongingToTheLabel$(F,label)
    for each $f \in V$ **do**
      $F_{ilabel} \leftarrow CountFrequencyOfFeatures(text_{label}, f)$
      for each $f \in V$ **do**
        $condProb[f][label] \leftarrow \frac{F_{ilabel}}{\sum_{i' \in V} F_{i'label}}$
      end for
    end for
  end for
  return $V, priorProb, condProb$
end procedure

procedure TESTINGNB(L, V, $priorProb$, $condProb$)
  $W \leftarrow ExtractFeatures$(V)
  for each $label \in L$ **do**
    $score_{label} \leftarrow logpriorProb[label]$
    for each $f \in W$ **do**
      $score_{label} += logcondProb[f][label]$
    end for
  end for
  return $argmax_{label \in L} score[label]$
end procedure

Figure 3. Naïve Bayes algorithm.

In the Figure 4 the system that is used for text classification is depicted and it is based on the Naïve Bayes approach. The system requires the labeled dataset as an input, which is later processed to extract features that are fed to the classifier. After classification performed, the system is tested to check how accurate results are.

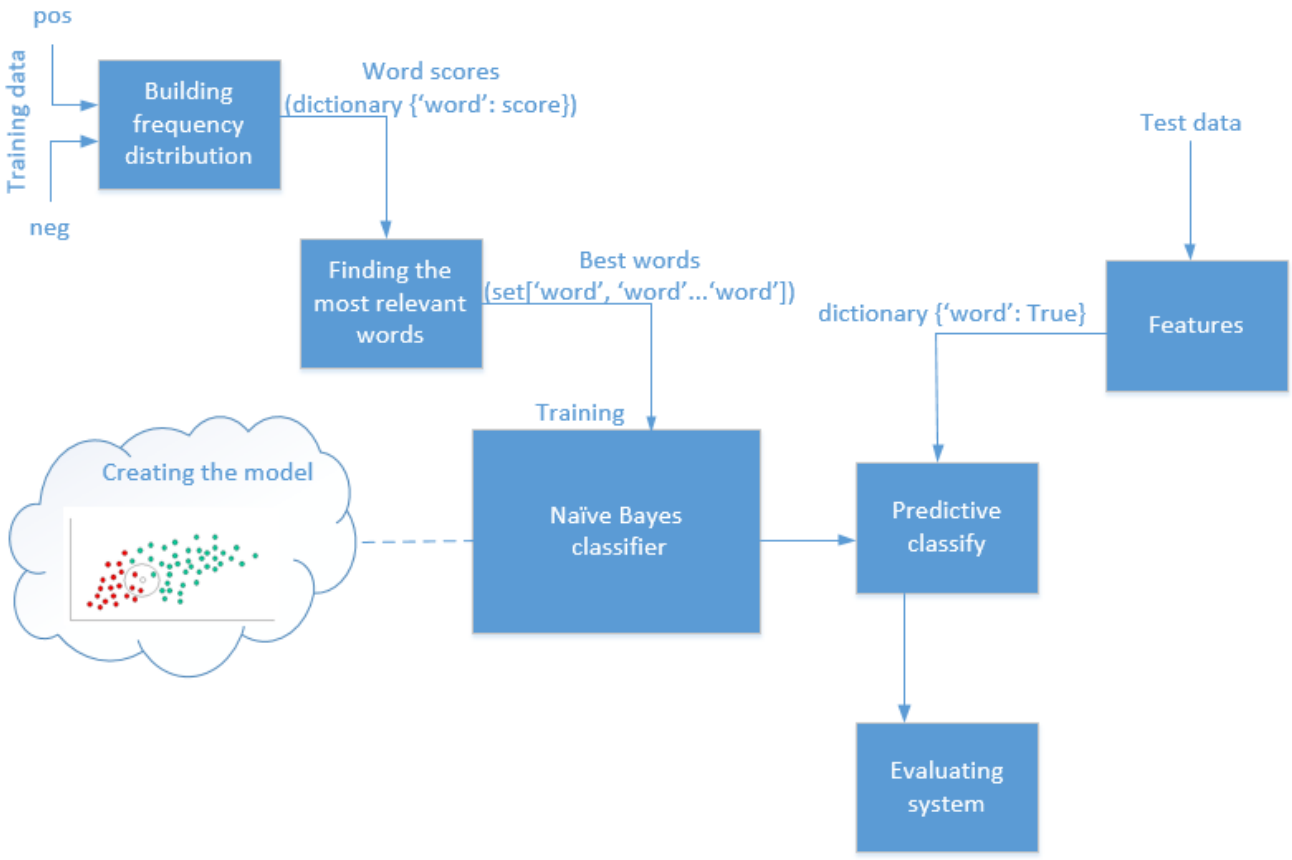

Figure 4 – Model for estimation the sentiment using Naïve Bayes approach.

3.3.3.2 Convolutional neural network (CNN)

Let's assume that transformation of the input text is performed and now it is represented as a high-dimensional vector. The next step is to apply convolution, non-linear and max-pooling operations to extract features from the input data. Below you can find explanations how extraction is happening on different layers of the CNN.

Convolutional layer. Convolutional layer allows retrieving patterns that are frequent in data [30] (it finds regions that are crucial for feature selection). More specifically in order to obtain a new feature c the convolution operation has to be applied. To that extend, n words from the

sentence are convolved with the filter weights *w* to obtain a feature map (see Figure 5). The size of the filter corresponds to the number of words we slide over (at this work filter size is 3, 4, 5). Filter weights are initialized randomly in the beginning and then adjusted through the training process. The feature can be mathematically represented as [32]:

$$c_i = f(w \cdot s_{i:i+n-1} + b)$$

where *w* is a vector of weights, " · " refers to the dot product, $s_{i:i+n-1}$ is a slide window, $b \in \mathbb{R}$ is a bias vector, and f is a non-linear function.

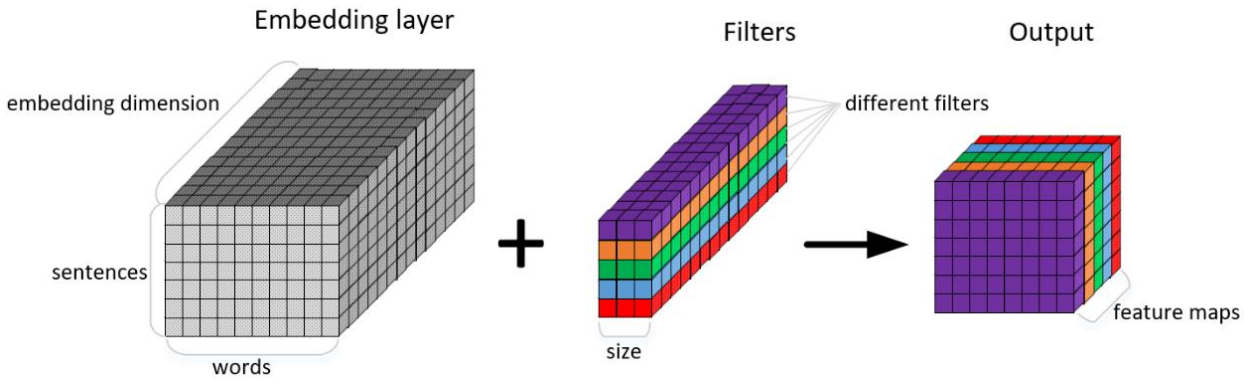

Figure 5. Convolution operation.

The filter is employed to each sequence of words in the sentence that corresponds to the filter size $\{s_{1:n}, s_{2:n+1}, \dots, s_{s-n+1:s}\}$ to generate a feature map [32]:

$$c(w) = [c_1, c_2, \dots, c_{s-n+1}]$$

Rectified Linear Unit (ReLU) is taken as a non-linear function as it is extensively used by researchers [30], [32], [41] and applied element-wise after convolutional layer. All negative values in the feature map converted to 0 to guarantee that the feature maps are positive (see Figure 6) [30]. ReLU allows producing a non-linear decision boundary. ReLU is defined as:

$$f(x) = \max(0, x)$$

Figure 6. ReLU activation function.

Pooling layer. After ReLU is applied to the convolutional layer it produces the input for the pooling layer. In this work max-pooling operation is performed that allows reducing the size of the feature map at the same time preserving the most relevant feature:

$$\hat{c} = \max \{c(w)\}$$

Such operation provides a single feature $\hat{c}$ for the feature map produced by the particular kernel *w*.

In this dissertation work, 128 filters are applied for each filter size that produces 128 feature maps respectively. The size of the latter depends on the following parameters: number of filters, stride that is defined by the number of unit by which kernel matrix is slide over the input, and zero-padding (adding zeros to the edges of the sentences ensures that words on the border will be processed by the filter, this operation is also known as wide convolution, if zero-padding is not applied that it is called narrow convolution) [40].

As mentioned above different filter sizes are applied. After max-pooling is performed using multiple filters, these outputs passed to the fully connected layer where they are concatenated into one feature vector. Using the latter softmax layer outputs the probability distribution over two classes (positive or negative) [30]:

$$p(y = j|x) = \frac{e^{x^T w_j + b_j}}{\sum_{k=1}^{K} e^{x^T w_k + b_k}}$$

where *x* is the output of the penultimate convolutional and pooling layers represented as a dense vector, w_k is a vector of weights of the *k*-th class, and b_k is bias of the *k*-th class.

Dropout is a way of preventing the network from overfitting. Training is usually performed using a stochastic gradient descent by randomly selecting some samples from the dataset. Dropout ensures regularization and applied before fully connected layer. Dropout method assumes that only on the training stage some portion of neurons is removed (dropout rate is set to 0.5) that prevents co-adaptation of neurons and leads to learning more robust features and makes model generalize new data well. Output after applying dropout is represented as:

$$y = w \cdot (z \circ r) + b$$

where penultimate layer $z = [\widehat{c_1}, \dots, \widehat{c_m}]$, "$\circ$" is the element-wise multiplication, and r is a vector containing 0s and 1s.

Overall, dropout technique speeds up the training.

CNN model is trained to minimize cross-entropy loss function that can be estimated as:

$$H(p, q) = -\sum_{x} p(x) \log q(x)$$

where $p(x)$ is the true probability (correct answer), $q(x)$ is the estimated probability.

Training of the CNN assumes the adjustment of the network parameters. This tuning process called backpropagation error. Backpropagation is applied to compute the gradient of the error function with respect to the filter weights. Adam algorithm [42] that is a stochastic gradient descent algorithm is used for optimizing parameters of the CNN (updating weights).

The model of convolutional neural network is depicted in Figure 7. Output sizes produced after each layer are given in figure below, where *batch* corresponds to the batch size and equals to 64; *len* corresponds to the maximum sequence length in the dataset; *dim* stands for embedding dimensionality and constitutes 128; *filter_size* is 3, 4, 5 respectively (depicted by three colors); *num_filters* is 128 and corresponds to the number of filters. Each filter slides over 128-dimensional embedding considering filter size. Stride size is equal to 1 (filter shift per step).

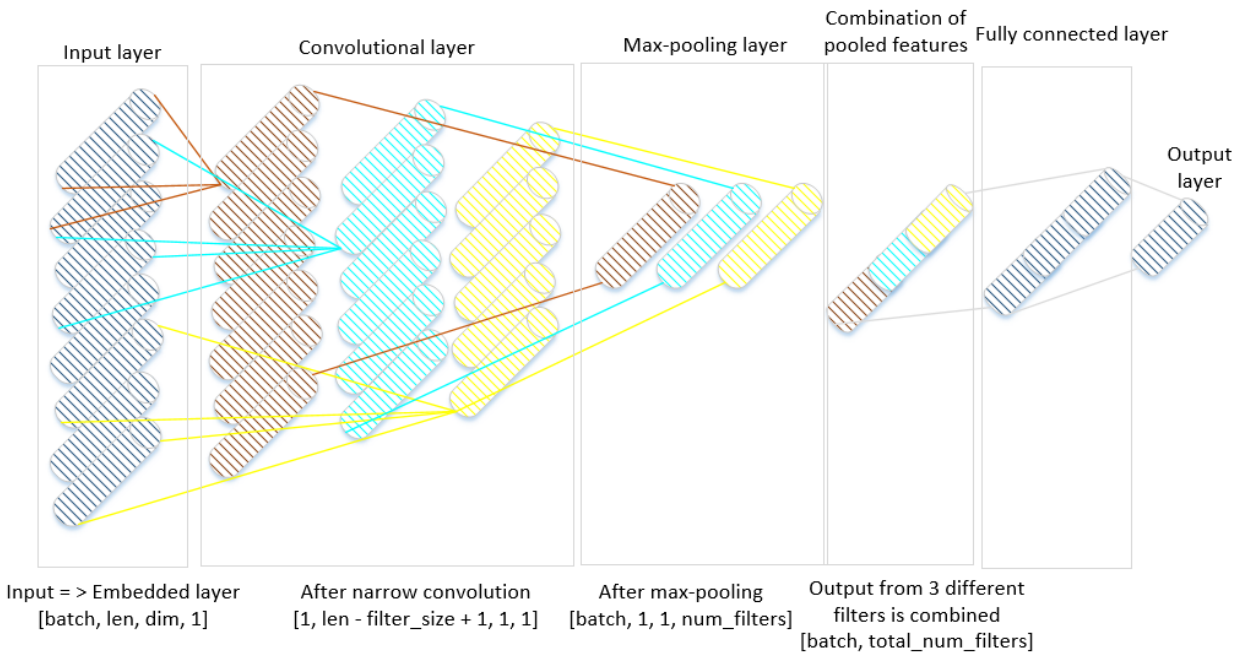

Figure 7. Architecture of the CNN.

3.4 Conclusion

Naïve Bayes approach is used quite often by researchers due to its simplicity and good performance despite its assumption about the independence of features. Moreover, neural networks became highly employed by scientists during last decade especially after Krizhevsky et al. [45] presented the results they obtained for image and video recognition task. Further studies showed that CNN is also applicable for Natural Language Processing tasks and can effectively classify text [32], [33].

The main part of this chapter illustrates details of the algorithms used for text classification, namely Naïve Bayes and convolutional neural network. Moreover, importance if preprocessing is described in the current chapter. The approaches for feature selection also introduced in this section.

4 Results and analysis

4.1 Introduction

This chapter introduces the results that were obtained after conducting the experiments using Naïve Bayes algorithm and convolutional neural network. The experiment is performed on two different datasets. The first dataset contains the movie reviews, second contains the tweets. Both datasets are labeled.

Naïve Bayes algorithm implemented using NLTK library and neural network using Tensorflow. Training and testing of the system were performed on the Rocket cluster that has 135 nodes (20 cores of 2.20 GHz, 64 GB RAM, 1 TB HDD) and helps to speed up the execution.
To evaluate the quality of the classification algorithms three main metrics are used, namely precision, recall, and F_1 score. Moreover, during training and testing stages, computational time was measured that is also used in the analysis of algorithms' performance.

Overall, the results are discussed. Based on the obtained information, the conclusion about the most efficient algorithm is given.

4.2 Evaluation metrics of algorithms performance

The effectiveness of the classification algorithms is usually estimated based on such metrics as precision, recall, F_1 score, and accuracy. Moreover, it is very important to take into account computational cost resources that algorithm needs for building the classifier and using it.

Consider the metrics that were used for calculation of the precision, recall, F_1 score, accuracy (see the Table 2). Confusion matrix contains the estimated and actual distribution of labels. Each column corresponds to the actual label and each row corresponds to the estimated label of the sentence.

Table 2. Confusion matrix for a binary classifier.

		Actual	
		positive	negative
Estimated	positive	TP	FP
	negative	FN	TN

TP is the number of true positives: the sentence that is actually positive and was estimated as positive,

TN is the number of true negatives: the sentence that is actually negative and was estimated as negative,

FP is the number of false positives: the sentence that is actually negative but estimated as positive,

FN is the number of false negatives: the sentence that is actually positive but estimated as negative.

Accuracy presents the proportion of the correct answers that are given by the classifier hence it can be estimated as:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision can be estimated using following formula:

$$precision = \frac{TP}{TP + FP}$$

Precision shows how many positive answers that received from the classifier are correct. The greater precision the less number of false hits. However, precision does not show whether all the correct answers are returned by the classifier. In order to take into account the latter recall is used:

$$recall = \frac{TP}{TP + FN}$$

Recall shows the ability of the classifier to “guess” as many positive answers as possible out of the expected.

The more precision and recall the better. However, simultaneous achievement of the high precision and recall is almost impossible in real life that is why the balance between two metrics has to be found. F_1 score is a harmonic mean of precision and recall:

$$F_1 = \frac{2 * precision * recall}{precision + recall}$$

4.3 Performance statistics

This subsection describes the conducted experiments and provides the results of the classification as well as evaluation criteria of the algorithms.

4.2.1 Naïve Bayes classifier

Naïve Bayes classifier was trained and tested on two datasets: movie reviews and tweets. For the Naïve Bayes classifier, all the experiment were conducted using the different amount of word for training the classifier, namely the *n* words that have the highest score were fed to the classifier. This score was calculated using χ^2 test, for this purpose frequency distribution of all words in the dataset was found as well as the conditional frequency is defined to count how many times a word has occurred in the positive sentence and how many times in the negative.

The first experiment involves the Naïve Bayes classifier which learned from movie reviews and evaluated on the movie reviews. The result of the first experiment is depicted in the chart below (see Figure 8).

It can be seen from the chart that on a small dataset (up to 500 words) all demonstrated metrics have lower values compared to the usage of the larger amount of words for training. However, it is also important to notice that as some point all metrics take the same value and then the decrease in values of all metrics can be observed. The highest accuracy is reached when 5000 informative words are taken as features and it constitutes 86,610%. Moreover, the classifier that is trained on 5000 of the best word also shows the highest values of recall and F_1 score. Recall equals to 86,704% and F_1 score is 86,623%. Nevertheless, the highest precision is gained when 6000 words are used for learning the classifier and makes up 86,907%. Note that is a case of sentiment classification precision is more important metric because the classifier has to be precise in

detecting true positive answers. Hence, the usage of 6000 words is most favorable for training the classifier on movie review in order to get the optimal performance in recognizing the positive and negative tweets.

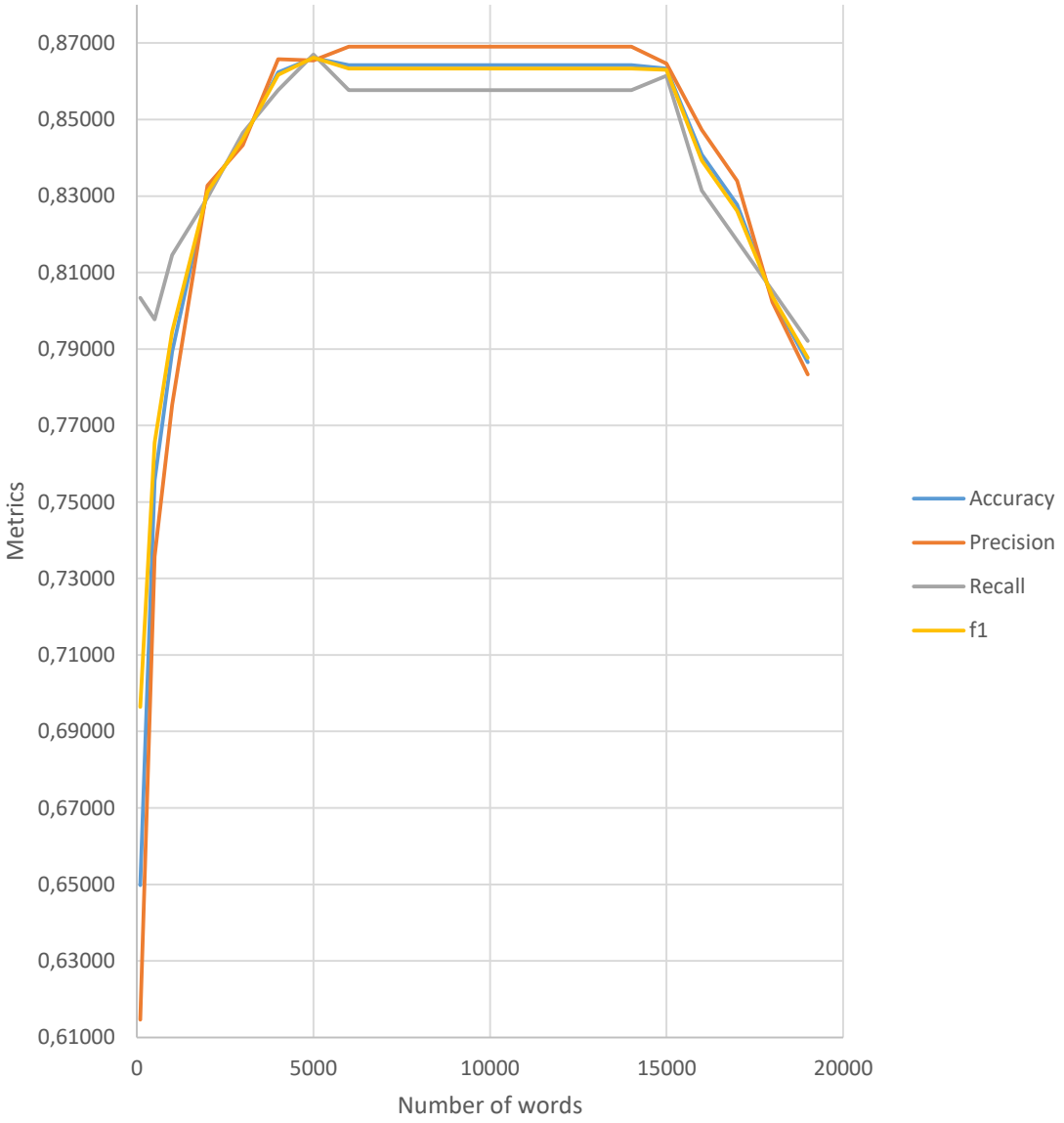

Figure 8. Evaluation of Naïve Bayes classifier that was trained on the movie reviews and tested on movie reviews (the values are specified in fractions).

The next test is performed using the same classifier that is trained on movie reviews, but evaluation is done on tweets. The metrics obtained after testing the classifier is illustrated in Figure 9.

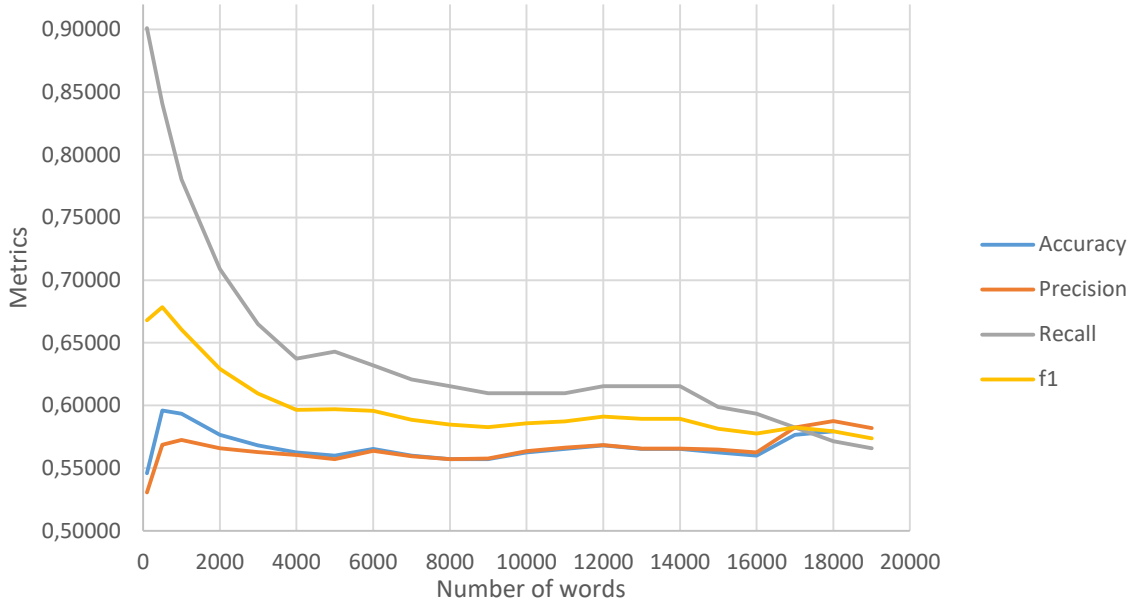

Figure 9. Evaluation of Naïve Bayes classifier that was trained on the movie reviews and tested on tweets (the values are specified in fractions).

The figure shows all the metrics got the lower values opposed to the previous case. The highest accuracy is reached when 500 words are used for training the classifier and it equals to 59,610%. Furthermore, F_1 score gets its optimal value of 67,849% if 500 words are used as features. However, the highest value of recall is gained when using only 100 words and it constitutes 90,11%. On the other hand, the optimal precision is reached when the classifier is learned from the whole dataset. Such situation happens because different data is used for training and testing the system. The context of the data used for training has a huge impact on the performance of the algorithm. As mentioned above, tweets differ from the usual sentences, such as reviews due to its informal lexicon that classifier does not know. Moreover, tweets may contain spelling mistakes, abbreviations, words elongation that are less often for reviews. Overall, the results show that if the classifier is trained on the movie reviews it performs better on classifying the movie reviews than classifying the tweets. The precision of the model classifying movie reviews is 27% higher than the precision of the one classifying the tweets.

The third experiment was conducted on the model that is trained on the larger dataset, which contains 1,6M tweets and tested on the tweets that were used for evaluation before. The result of the evaluation is depicted in Figure 10.

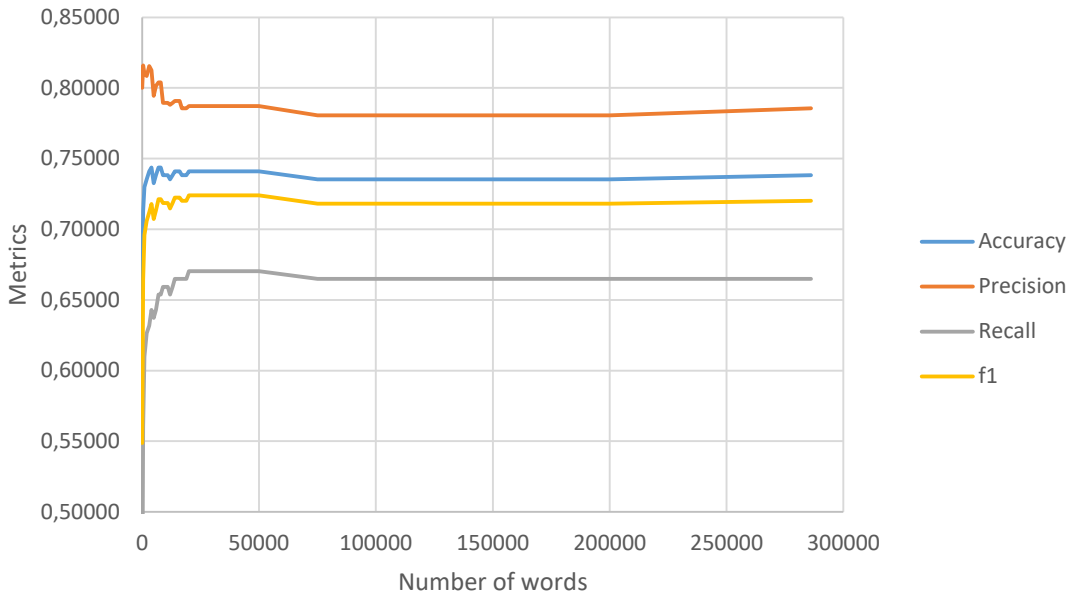

Figure 10. Evaluation of Naïve Bayes classifier that was trained on the tweets and tested on tweets (the values are specified in fractions).

From the figure, it can be seen that the classifier that is trained on the tweets classifies tweets better than the one that is trained on the movie reviews. The maximum of the accuracy is achieved when classifier takes 4000 words as features for learning and the accuracy constitutes 74,373%. On the other hand, the highest values of the recall and F_1 score are reached when the number of features makes up 20000 words and equal to 67,033% and 72,404% respectively. However, the optimum in the precision can be obtained if 3000 features used for training. The precision of the model that is trained and tested on tweets is 22% higher than the precision of the one that is trained on movie reviews but tested on tweets and constitutes 81,5%.

To sum up, when the classifier is trained and tested on the same type of data it shows better performance. Moreover, it has been found that the classification model that is based on the Naïve Bayes approach does not require huge training dataset, however, it needs the data samples from the same domain for training and testing the classifier.

Furthermore, computational cost is estimated. More specifically, during the training process that includes preprocessing and feature selection, the usage of virtual memory resource was evaluated. Figure 11 illustrates the memory usage progress while training the Naïve Bayes classifier on reviews.

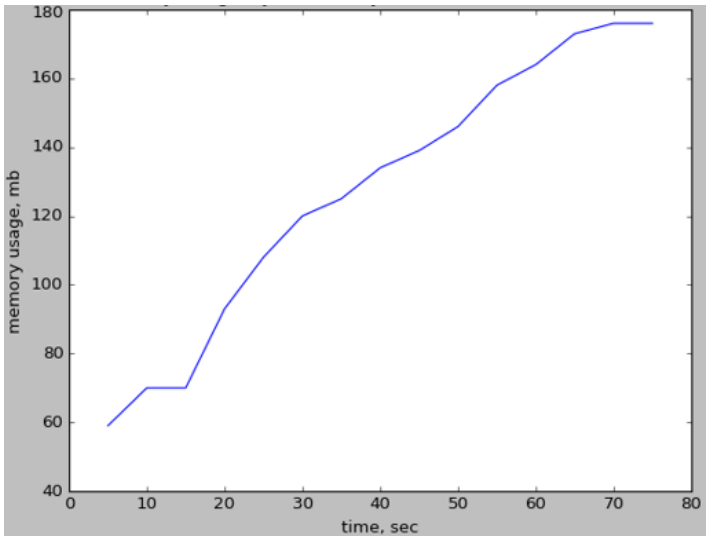

Figure 11. Memory usage when building Naïve Bayes classifier that is learned from movie reviews.

The chart indicates that training process of the classifier takes a bit more than a minute and requires less than 180 MB of memory. Hence, it shows that training process is fast.

The same dependency was retrieved for the classifier that is trained on tweets (see Figure 12). The increase of the memory usage, as well as growth of the training time, are represented in the chart below. This time the classifier is more memory demanding and it consumes around 3,5 GB of memory. Moreover, the time spent on training also rose significantly and constituted around 50 min for the case when NB model is trained on tweets. Such growth can be explained by the employment of much larger dataset that needs more powerful computational resources as opposed to the previous model. Part of the chart that is a straight line reflects preprocessing and feature selection, next the growth of memory is observed when classifier learns.

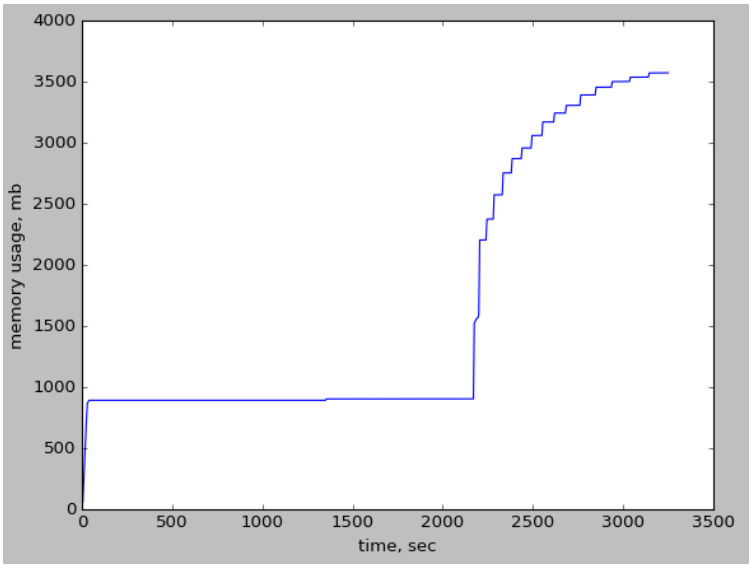

Figure 12. Memory usage when building Naïve Bayes classifier that is learned from tweets.

4.2.2 Convolutional neural network

The following experiments are performed by employment of the convolutional neural network that has one layer and uses the randomly initialized word embeddings that are convolved with 3 different filter sizes. In the first experiment, the CNN was trained on the movie reviews and tested on tweets. Results are presented in Table 3.

Table 3. Evaluation of the CNN that is trained on the movie reviews (the values are specified in fractions).

	accuracy	precision	recall	f1
CNN movie reviews	0,599	0,623	0,527	0,571

The accuracy is 59,9% that is a bit better that what was obtained using the Naïve Bayes classifier (trained on movie reviews and tested on tweets). Therefore, the accuracy is 1,4% higher opposed to Naïve Bayes. However, the recall and F_1 that are calculated based on CNN model show worse output. To be precise, the recall constitutes 52,7% that is 4,4% less opposed to the recall that is gained using corresponding Naive Bayes classifier, score makes up 57,1% that is 1,2% less

compared to the same Naïve Bayes model. The precision has 2,87% rise applying CNN against NB.

CNN did not show great performance on movie review dataset, because usually neural network requires larger dataset (millions of samples) for training. Hence, it is not enough data for the model to generalize well an unseen samples that leads to such insignificant results that CNN produced.

In addition, the CNN is trained on tweets. Later the system is evaluated on tweets and results are introduced in table below.

Table 4. Evaluation of the CNN that is trained on the tweets (the values are specified in fractions).

	accuracy	precision	recall	f1
CNN tweets	0,791	0,761	0,857	0,806

CNN model shows a 5,28% increase in accuracy compared to the Naïve Bayes classifier and it makes up 79,1%. The growth of the recall and F_1 score are also observed and they constitute 85,5% and 80,6% respectively. Hence, an improvement of recall is almost 20% and F_1 score enhancement is almost 9%. However, the slight decrease of precision is demonstrated by CNN classifier, in this case precision is 76,1%.

Comparison of CNN performance is made with the results of the Naive Bayes classifier that is trained on all words from movie review dataset.

As mentioned earlier, memory usage is estimated and used as an additional metric for assessment of the classifier performance. This metrics refers to the computational cost that is spent on building and using the classification model. Consider Figure 13, which plots the memory usage against the time for the CNN classifier that is learned from movie reviews.

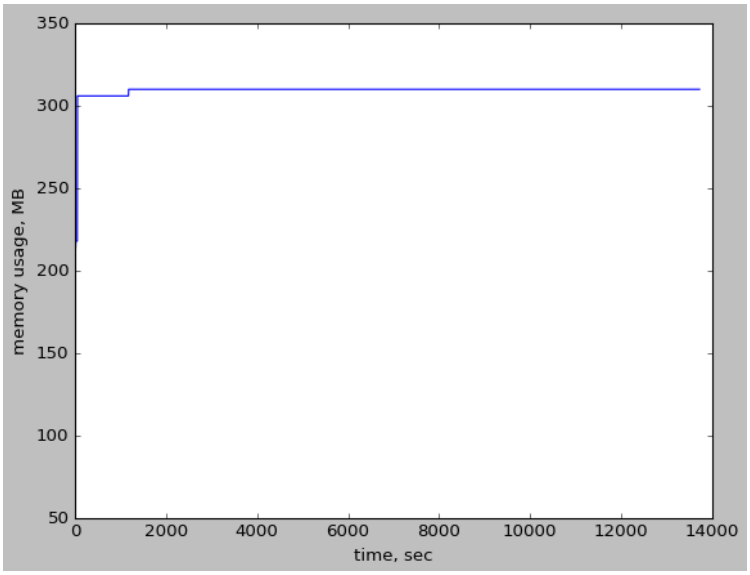

Figure 13. Memory usage when building CNN classifier that is learned from reviews.

It can be seen from the chart that such model requires 310 MB, that is almost twice higher that NB classifier needs for the respective dataset. However, the computational time has increased to almost 4 hours.

The next chart represents the same dependency but for classifier that is trained on tweets (see Figure 14).

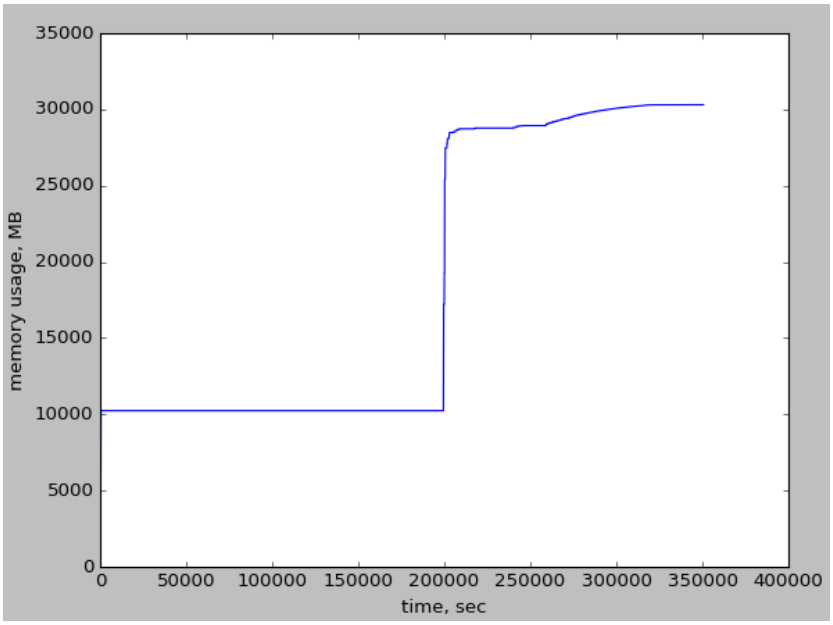

Figure 14. Memory usage when building CNN classifier that is learned from tweets.

The chart illustrates that the classifier that is trained on a huge dataset and consumes a lot of memory that is around 30 GB. Furthermore, the computational time has increased from several hours to several days. All these factors make the CNN training expensive.

Classifier was evaluated on tweets that were labeled before. Moreover, 122 tweets were retrieved from the Twitter based on the query "eurovision" in order to demonstrate its performance. Tweets were collected during 5 days, one call per day was done to the Twitter API, then they were manually filtered to exclude neutral tweets, because classifier was not trained on such type of tweets. In Table 5, example of the estimated sentiment is given. Label that is equal to 1 corresponds to the positive sentiment, 0 – negative. Most of the labels were correctly assigned, but some of them got the wrong label.

Table 5. Example of tweet categorization.

text	label
i am so excited for eurovision this saturday such a brilliant few hours of television	1
sorry but this isn't love this is shit	0
eurovision has only just started and I'm already exhausted of all the different emotions	0
damn I hate scott mills better keep him far away from eurovision next time	0
i love the eurovision song contest	1
this guy understands music and brought authenticity to the stage stunning performance and song love it	1
australia is so good on stage love	1
i don't have poland high enough on my final top 42 22nd isn't high enough sorry top 15 at least	1
wtf almost all eurovision songs i've enjoyed didn't make it to the final	1
the perfect ending so much fun so much music thank you and goodnight	1

The histogram (see Figure 15) illustrates the volume of tweets (positive and negative) throughout the five days. It can be seen that majority of tweets have positive polarity, that means people like the Eurovision show in general.

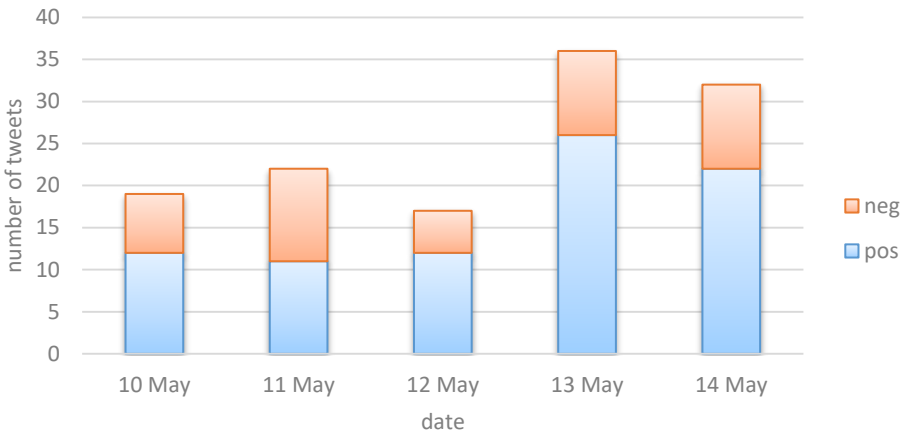

Figure 15. Volume of tweets over time.

4.4 Conclusion

This chapter presents the results of conducted experiments using Naïve Bayes and CNN classifiers. It can be observed that Naïve Bayes approach gave quite good results. Nonetheless, CNN outperforms the Naïve Bayes a bit (see Figure 16). As mentioned above, when dealing with sentiment classification task, the precision is the metric that has to be high in order to define true sentiment expressed in the sentence, in this case, recall can deteriorate.

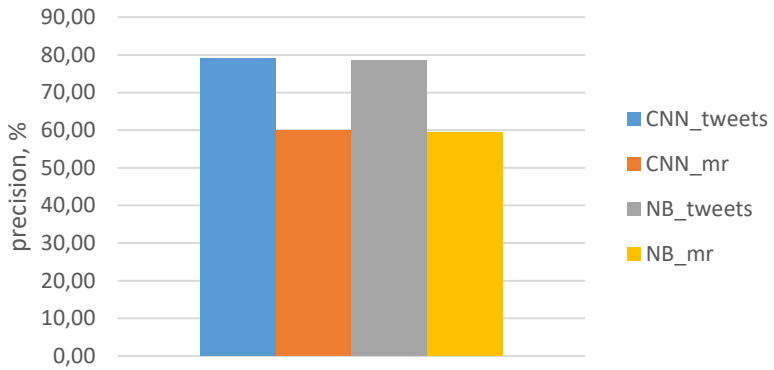

Figure 16. Precision of different classifiers.

Moreover, the behavior of the classifier training process was investigated, namely, it was estimated how much memory the classifiers consume to be trained on different sizes of the datasets. It is important to notice that CNN requires way more memory than NB. However, CNN

classifier produces similar metrics as NB. Therefore, analysis of the results shows that investigated models may be further improved because metrics of the accuracy, precision, recall and F_1 score are not significant as they were expected, especially when employing CNN classifier. It is observed that Twitter data is noisier opposed to the movie reviews that make classification of tweets more difficult. Some preprocessing techniques were applied to the tweets (see detailed explanations in Section 3.3.1), but it seems that more sophisticated methods should be used for filtering the tweets from the noise.

To conclude, the classifier that is based on the Naïve Bayes approach has shown comparable results with CNN classifier, despite its simplicity, also it is less resource-demanding opposed to the CNN model. In general, CNN should perform better, but CNN need much larger dataset to be fed to the classifier. The figure below illustrates how precision metric of the CNN classifier can be affected by the number of training sample. The fluctuation can be seen in the chart (Figure 16) which is due to the adjustment process of the weights assigned to the filters; however, more instances are given to the classifier better performance is achieved. This indicates that CNN requires large dataset for training in order to give good results. However, collecting and labeling huge dataset requires a lot of time which we did not have unfortunately during this thesis work period.

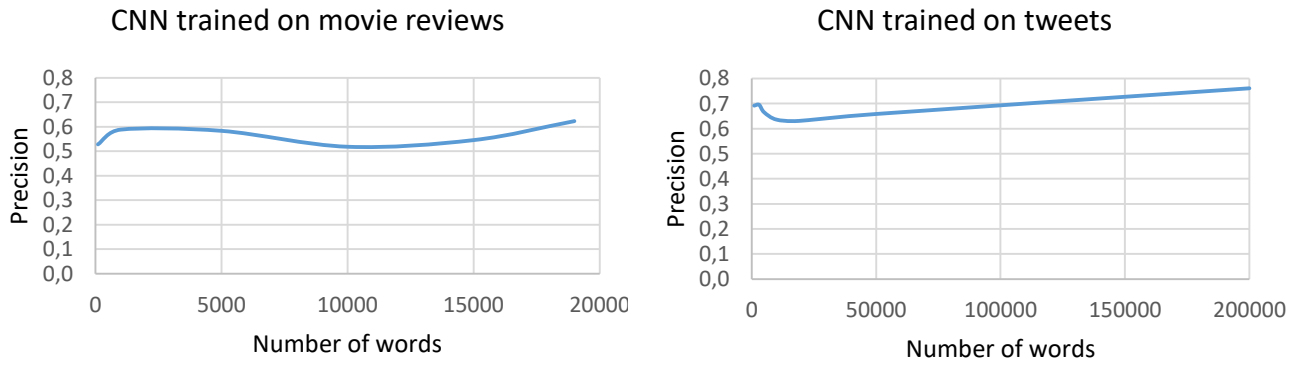

Figure 16. Dependency of precision versus number of word feed to the classifier

In addition, it was investigated that the context of the dataset highly affects the performance of the classifier. If the task is to classify the data from whatever domain, then the classifier has to know samples that capture varied context. Dataset has to contain such types of data as tweets, reviews, news from different domains such as science, politics, and economy. The classifier that

is trained on the diverse data with different context will highly probably be able to detect correct sentiment when it is tested across all domains. Hence, the quality of the dataset has an enormous impact on the effectiveness of the classification model.

5 Conclusion

This chapter introduces conclusions of the work that was done as well as discussions about possibilities for performing future work.

5.1 Conclusion

Sentiment analysis task is under research since the early 2000s and it is still in developing phase, especially the exploration of microblogs, such as Twitter. Twitter message is less informative opposed to usual review or comment and also contains a lot of noisy data that makes classification of tweets more challenging.

This dissertation investigates the algorithms that can be used for sentiment classification. According to the literature review, it was found that majority of sentiment analysis approaches on tweets rely on supervised machine-learning methods. Therefore, it was decided to study Naïve Bayes and Convolutional neural network approaches as far as these methods are in trend among researchers and they provide meaningful results. Hence, analysis of both algorithms was carried out and their performance was estimated. The classification model was trained on two different datasets in order to study whether sentiment classification is the domain-dependent task or not. Additionally, two feature models were investigated, more specifically, unigrams are used for training the Naïve Bayes classifier and n-grams are employed for CNN classifier. Furthermore, the importance of the preprocessing stage when tweets are utilized as training data is discussed.

In this thesis binary classification is considered, namely, the tweet/review is assigned a positive or negative label according to the sentiment conveyed in it. Two different classifiers were investigated in order to estimate the sentiment. Classifiers performance is evaluated based on experiments. The first supervised method that was explored in this dissertation is Naïve Bayes approach. As was expected it has shown sufficient results on the tweet classification. The best result of the precision that was achieved, made up 78,57% when NB classifier was learned from the whole set of tweets. Another supervised approach that was studied for training the classifier is the one-layer convolutional neural network. After evaluation of the CNN, the precision has slight growth and constituted 79,10%. However, it was discovered that the CNN is extremely resource-

demanding opposed to NB. In general, CNN performs better that Naive Bayes classifier, but it requires solid computational resources and large amount of training sample.

Additionally, this study has shown that in order to achieve meaningful performance of the classifier it has to be trained and tested on the same type of the dataset because the correlation exists between the classifier performance and domains, which are used for collecting training and testing samples. Moreover, it was observed that usage of n-grams versus unigrams has slightly improved the efficiency of the classification model.

The recommendations that can be applied for the model to improve the performance of the classifier are described in the next section. The recommendations need to be further checked that is why they are introduced in the subsection that is future work.

5.2 Future perspectives

Future work will involve investigation of other approaches for preprocessing tweets because they have to be more thoroughly filtered to achieve the higher accuracy, precision, etc. There are several directions that can be performed:

- As mentioned earlier, tweets may contain a lot of spelling mistakes, hence, spelling corrector can be applied to exclude typos.
- Additionally, tweets contain huge amount of emoticons and expressions that convey laugh, such as lol, ha-ha-ha, jaja that have to be generalized and labeled whether emoticon/expression refers to a positive or negative meaning, the ones that are ambiguous (e.g. emoticon with stuck-out tongue " :-P ") have to be removed from the training dataset.
- Another experiment that may be carried out is the replacement of the abbreviations with their full meaning. It obviously will increase the size of the training corpus but may add more sense to the tweet.
- Moreover, it would be interesting to add neutral class and check the performance of the classifier. However, in this case, the training and testing datasets have to include neutral samples to feed the model and evaluate it.

References

[1] Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, *5*(1), 1-167.

[2] Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, *5*(4), 1093-1113.

[3] Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, *1*(12).

[4] Bütow, F., Schultze, F., & Strauch, L. Semantic Search: Sentiment Analysis with Machine Learning Algorithms on German News.

[5] Pak, A., & Paroubek, P. (2010, May). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In *LREc* (Vol. 10, No. 2010).

[6] Gokulakrishnan, B., Priyanthan, P., Ragavan, T., Prasath, N., & Perera, A. (2012, December). Opinion mining and sentiment analysis on a twitter data stream. In *Advances in ICT for emerging regions (ICTer), 2012 International Conference on* (pp. 182-188). IEEE.

[7] Hallsmar, F., & Palm, J. (2016). Multi-class sentiment classification on twitter using an emoji training heuristic.

[8] Turney, P. D. (2002, July). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 417-424). Association for Computational Linguistics.

[9] Salas-Zárate, M. D. P., Medina-Moreira, J., Lagos-Ortiz, K., Luna-Aveiga, H., Rodríguez-García, M. Á., & Valencia-García, R. (2017). Sentiment Analysis on Tweets about Diabetes: An Aspect-Level Approach. *Computational and mathematical methods in medicine, 2017*.

[10] Chiavetta, F., Bosco, G. L., & Pilato, G. (2016). A Lexicon-based Approach for Sentiment Classification of Amazon Books Reviews in Italian Language.

[11] Hailong, Z., Wenyan, G., & Bo, J. (2014, September). Machine learning and lexicon based methods for sentiment classification: A survey. In *Web Information System and Application Conference (WISA), 2014 11th* (pp. 262-265). IEEE.

[12] Hu, M., & Liu, B. (2004, August). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 168-177). ACM.

[13] Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, *38*(11), 39-41.

[14] Dong, Z., Dong, Q., & Hao, C. (2010, August). Hownet and its computation of meaning. In *Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations* (pp. 53-56). Association for Computational Linguistics.

[15] Musto, C., Semeraro, G., & Polignano, M. (2014). A comparison of lexicon-based approaches for sentiment analysis of microblog posts. *Information Filtering and Retrieval*, *59*.

[16] Kim, S. M., & Hovy, E. (2004, August). Determining the sentiment of opinions. In *Proceedings of the 20th international conference on Computational Linguistics* (p. 1367). Association for Computational Linguistics.

[17] Park, S., & Kim, Y. (2016, June). Building thesaurus lexicon using dictionary-based approach for sentiment classification. In *Software Engineering Research, Management and Applications (SERA), 2016 IEEE 14th International Conference on* (pp. 39-44). IEEE.

[18] Hatzivassiloglou, V., & McKeown, K. R. (1997, July). Predicting the semantic orientation of adjectives. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics* (pp. 174-181). Association for Computational Linguistics.

[19] Ding, X., Liu, B., & Yu, P. S. (2008, February). A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 international conference on web search and data mining* (pp. 231-240). ACM.

[20] Thakkar, H., & Patel, D. (2015). Approaches for sentiment analysis on twitter: A state-of-art study. *arXiv preprint arXiv:1512.01043*.

[21] Turney, P. D. (2002, July). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 417-424). Association for Computational Linguistics.

[22] Rothfels, J., & Tibshirani, J. (2010). Unsupervised sentiment classification of English movie reviews using automatic selection of positive and negative sentiment items. *CS224N-Final Project*.

[23] Zagibalov, T., & Carroll, J. (2008, August). Automatic seed word selection for unsupervised sentiment classification of Chinese text. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1* (pp. 1073-1080). Association for Computational Linguistics.

[24] Tang, B., Kay, S., & He, H. (2016). Toward optimal feature selection in naive Bayes for text categorization. *IEEE Transactions on Knowledge and Data Engineering*, *28*(9), 2508-2521.

[25] Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, *1*(12).

[26] Gautam, G., & Yadav, D. (2014, August). Sentiment analysis of twitter data using machine learning approaches and semantic analysis. In *Contemporary computing (IC3), 2014 seventh international conference on* (pp. 437-442). IEEE.

[27] Pang, B., Lee, L., & Vaithyanathan, S. (2002, July). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10* (pp. 79-86). Association for Computational Linguistics.

[28] Chikersal, P., Poria, S., & Cambria, E. (2015, June). SeNTU: sentiment analysis of tweets by combining a rule-based classifier with supervised learning. In *Proceedings of the International Workshop on Semantic Evaluation, SemEval* (pp. 647-651).

[29] Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval* (Vol. 1, No. 1, p. 496). Cambridge: Cambridge university press.

[30] Severyn, A., & Moschitti, A. (2015, August). Twitter sentiment analysis with deep convolutional neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 959-962). ACM.

[31] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111-3119).

[32] Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

[33] Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

[34] Britz, D. (2015). Recurrent Neural Networks Tutorial, Part 1–Introduction to RNNs. [WWW] http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/ (05.05.2017)

[35] Liu, P., Qiu, X., & Huang, X. (2016). Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*.

[36] Aggarwal, C. C., & Zhai, C. (2012). A survey of text classification algorithms. In *Mining text data* (pp. 163-222). Springer US.

[37] Pang, B., Lee, L. (2005). Movie Review Data. Sentence polarity dataset v1.0. [WWW] https://www.cs.cornell.edu/people/pabo/movie-review-data/ (05.05.2017)

[38] Go, A., Huang, L., & Bhayani, R. (2009). Twitter sentiment dataset. [WWW] http://cs.stanford.edu/people/alecmgo/trainingandtestdata.zip (05.05.2017)

[39] Go, A., Huang, L., & Bhayani, R. (2009). Twitter sentiment analysis. [WWW] http://hughchristensen.co.uk/papers/socialNetworking/Twitter%20Sentiment%20Analysis.pdf (05.05.2017)

[40] Karn, U. (2016). An Intuitive Explanation of Convolutional Neural Networks. [WWW] https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/ (05.05.2017)

[41] de Freitas Junior, E. (2016). A robust deep convolutional neural network model for text categorization. [WWW] https://www.dcc.ufmg.br/pos/cursos/defesas/1899M.PDF (05.05.2017)

[42] Kingma, D., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[43] Shepelenko, O. (2016) Sentiment analysis on tweets using ClowdFlows platform. [WWW] http://ds.cs.ut.ee/courses/course-files/report.pdf (05.05.2017)

[44] Chen, Y. (2015). Convolutional neural network for sentence classification. [WWW] https://pdfs.semanticscholar.org/6d35/949dac3d64ce087c54d200463b3908932832.pdf (05.05.2017)

[45] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

License

Non-exclusive licence to reproduce thesis and make thesis public

I, **Olha Shepelenko**,

(*author's name*)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:
   1.1. reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and
   1.2. make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

of my thesis

Opinion Mining and Sentiment Analysis using Bayesian and Neural Networks Approaches,

(title of thesis)

supervised by Amnir Hadachi and Artjom Lind,

(supervisor's name)

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, **18.05.2017**