Domain Independence of Machine Learning and Lexicon Based Methods in Sentiment Analysis.

Meriton Xhymshiti
University of Twente
Waalstraat 103, 7523 RC Enschede
the Netherlands
m.xhymshiti@student.utwente
.nl

1. ABSTRACT

Sentiment analysis is a sub-area in the field of Natural Language Processing (NLP) and it aims at automatically detecting the polarity of an opinion expressed on a textual information. There are two main approaches for analyzing a sentiment and determining its polarity: Lexicon based approaches and Machine Learning approaches. A lexicon-based approach uses a dictionary of words together with a polarity label for each of these words to determine the sentiment polarity of a document (e.g positive, negative or neutral). A machine learning approach trains a classifier in a labelled dataset and predicts sentiments using the model it creates. This paper presents a comparison on the domain independence of a ML system and lexicon-based system for Dutch sentiment analysis. The main contribution of this paper is that we show that in absence of "good-quality" labelled data for training in a specific domain, a lexicon-based system can be as good as a ML system. The dataset that will be used is in Dutch language and consists of large datasets of product and clothing reviews crawled from bol.com and a small dataset of "life memories" of people collected by researchers at the University of Tilburg. Pattern will be used as a lexicon based method and Support Vector Machines as a machine learning method.

2. KEYWORDS

sentiment analysis, domain-independence, machine learning method, lexicon-based method, support vector machine, pattern.

3. INTRODUCTION

With the rapid development of Web 2.0, people have begun to express their opinions on entities (e.g products, organization, people etc.) over the internet. Companies use sentiment analysis to gain more information about their product. They monitor social media mentions of their brands, determine the sentiment of the mass toward the brand and quickly respond to it. The user generated content on the internet covers different topics and most of the time expresses opinions of the mass, therefore being able to mine and analyze this information has been proven to be very beneficial to the industrial and academic community. Merchants selling products on the internet give the possibility to users to provide feedback about a certain product. Furthermore, social media companies, such as Twitter, offer the possibility to the users to express their opinions toward entities. The amount of opinions which is expressed over the internet nowadays is huge. A product can have hundreds or even thousands of reviews. This makes it difficult for the customer to read them and to conclude whether to purchase the product. It also makes it difficult for the manufacturer of the product to gain information about the opinion of others toward the product and whether improvements are needed. Sentiment analysis (also known as opinion mining) is a sub-area in the field of Natural Language Processing (NLP) and it aims at automatically detecting the polarity (e.g positive or negative) of an opinion expressed on a textual information. Recently, research on sentiment analysis has been conducted on product reviews, e-commerce and social media content [5,8,15].

There are two main approaches to analyzing a sentiment and determining the polarity, whether it expresses positive, negative or neutral opinion: rule-based (lexicon-based) approaches and machine learning approaches. A former uses a dictionary of opinion words (e.g "good" or "bad") to predict the polarity of a text document [16]. Each of these words are annotated with a polarity value, e.g "good" is annotated as a positive word. A latter trains a text classifier on a human labelled training dataset and predicts sentiments using the model it creates [3].

One fundamental problem in sentiment analysis is that a word conveys different polarity when used across different domains. For instance, the word 'horror' in a movie review does not mean that the review expresses a negative opinion towards the movie. On the other hand, if 'horror' is used in a product review, that usually implies a negative opinion towards that product. Furthermore sentiments are often expressed differently across different domains. A lexicon-based method needs to have an implementation that distinguishes words based on their context of use. On the other hand a machine learning system needs to have a large amount of labelled data on that specific domain, in order to perform well. Furthermore, research in the past has shown that if a machine learning system that is trained in an X domain, is used for predicting opinions from domain Y, the performance will not be satisfying [2]. Therefore, the system needs to be re-trained on that specific domain again, in order for it to perform well. However, for some languages like Dutch re-training is not possible, because there exists a large labelled dataset for product reviews, but for other domains such as Newspaper articles, there is no big annotated dataset. Therefore, the goal of this paper is to find out whether a rule-based system can be a better option than a ML classifier that has been trained in a domain consisting of a large annotated dataset and used for predicting sentiments outside the training domain. The research question that the paper aims to answer is the following:

- Is a lexicon-based system more domain independent than a machine learning system?

To address this a cross-domain comparison between a rule-based system and a ML system will be performed. Pattern [14] will be used as a rule-based system and Support Vector Machine [1] as a machine learning system. The dataset that will be used is in Dutch language and includes a large annotated dataset of product and clothing reviews collected from bol.com and a small annotated dataset of "life memories" collected by researchers at the University of Tilburg.

4. RELATED WORK

Research in the past has been conducted on classifying text by using a rule-based approach. In [7] Hu et al. performed a sentiment classification on product reviews. They manually create a small lexicon of seed adjectives tagged with positive or negative labels and then extend this lexicon using WordNet, which is an open source lexical database. Afterwards, they use this lexicon to summarize the number of positive and negative reviews of a product based on its features. In [6] Das and Chen use a manually created lexicon with some scoring functions to classify stock postings on an investor bulletin. Besides the work on lexicon-based approach, machine learning approach has also been applied in many experiments with regards to sentiment classification. In [3] Pang et al. conducted an experiment on sentiment classification by using multiple machine learning algorithms, such as Naive Bayes and Support Vector Machines. They use a large annotated dataset of movie reviews and train and test the classifiers within the movie domain. The experiment results showed that the SVM achieved a higher accuracy compared to other ML algorithms. In [15] Xing et al. performed sentiment analysis using product review data. They use different machine learning algorithms such as SVM, Naive Bayes and Random Forest to classify the product reviews into positive, neutral or negative classes.

In the past, research has been conducted on comparing lexicon-based and machine-learning methods. In [17] Hailong et al. perform a survey in cross-domain and cross-lingual comparison between lexicon-based methods and machine learning methods. In the survey they use SentiWordNet as a lexicon-based method and Naive Bayes, Support Vector Machines as machine-learning methods. After conducting the experiments, they came to a conclusion that supervised machine learning methods have higher precision compared to lexicon-based methods across different domains. However, The work on my paper differs from theirs in three main aspects. First, the focus of this paper is on sentiment analysis of dutch textual information, whereas they are applying sentiment analysis on english textual information. Second, they use SentiWordNet as a lexicon-based method, whereas in this paper Pattern will be used instead. Pattern takes into account negation and intensifiers when determining the polarity of a document (e.g "not good" and "very good"), whereas SentiWordNet does not. Third, their cross-domain analysis differs from the analysis on this paper, because the performance of the ML-based systems is always calculated after re-training the model on that specific domain and then the comparison is made with the lexicon-based systems. In this paper the ML-based system will be trained in one specific domain and then will be used for predicting the polarity of sentiments from other domains. Afterwards, the performance of it will be compared to the performance of Pattern. Mukhtal et al. conducted a comparison between a lexicon-based approach and a machine learning approach [11]. In contrast to the experimental results of Hailong et al [17], the lexicon-based approach outperformed the machine learning approach.

Research has been conducted on the domain-independence of machine learning systems. In [2] Aue et al. perform an experiment on the domain independence of the SVM classifier. They train the SVM classifier in movie domain and use it for predicting sentiments from the same domain and three other domains. The experimental results show that the classifier performs best when trained and tested on the same domain and the performance drops when the classifier is used to predict sentiments outside the domain. However, in that study there is no comparison with a lexicon-based method. The experimental results show that the performance of the machine classifier drops when the domain is changed, but it cannot be concluded that the performance will be worse or better than the performance of a lexicon-based method.

5. METHODOLOGY

In this section we present the methodology used in this experiment. The section is organized as follows: first we describe the datasets used and how they were pre-processed (5.1 and 5.2); then we will introduce the two systems that will be compared, namely the lexicon-based Pattern (5.3), and Support Vector Machines (5.4) representing a ML system; Afterwards we will present the metrics that has been used for the evaluation (6) and the results of the experiment (7); Finally, we will discuss the results and draw the conclusions on domain independence and relative performance of the systems (8).

5.1 Data collection

Data that will be used in the research includes clothing reviews, music reviews and life memories. Clothing and music review data were collected from bol.com, whereas life memories data were collected from the University of Tilburg. The researchers collected different life memories from a group of people and annotated the polarity for each memory. Each of these collected documents includes the textual opinion toward the entity/memory and a rating scale of one to five for the reviews

and one to seven for the life memories. A rating can be seen as a ground truth tag to the document, and will be used to evaluate the performance of the sentiment analysis systems that will be used in this experiment. The size of each dataset and the rating scale used is shown in Table 1.

Table 1. Dataset details.

Data type	Rating scale	Original size	Size after cleansing	Size after balancing
Clothing reviews	1-5	48233	17774	2420
Music reviews	1-5	64879	64874	4270
Life memories	1-7	120	120	120

5.2 Data Preprocessing

5.2.1 Data cleansing

Since part of the dataset that will be used is crawled from the web, it is needed to do some data cleansing before we start with sentiment analysis. The preprocessing step includes removing null values, duplicate values and html tags. After data cleansing has been performed in the bol dataset, the amount of reviews decreased significantly (see Table 1). Note that cleansing has not been applied to the life memories corpus, since the memories were manually collected by the researchers.

5.2.2 Data balance

After the data was cleansed, a large difference was noticed between the number of reviews per category in clothing and music corpus, especially between the number of one star reviews and five star reviews (see Figure 2&3). Therefore a decision has been made to balance the reviews per star rating.

This is necessary to avoid any favor of the machine classifier toward the majority class. To balance the dataset, we used a widely known technique: under-sampling. Under-sampling consists of removing samples from the majority class and equal the number of them to the number of samples of the minority class. After balancing, there were a total of 2420 clothing reviews (484 per class) and a total of 4270 music reviews (854 per class). Data balance has not been applied to life memory corpus since there were only 120 samples in total and thus there was no significant difference between the number of samples per class.

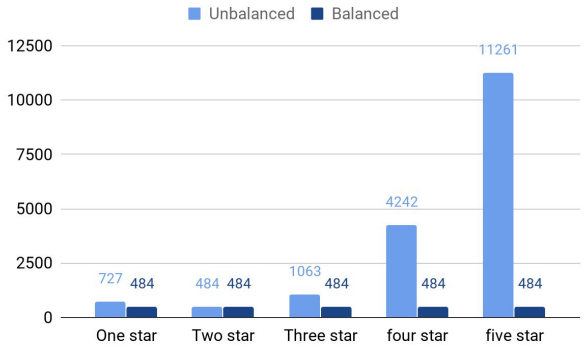

Figure 2. Class distribution of clothing reviews.

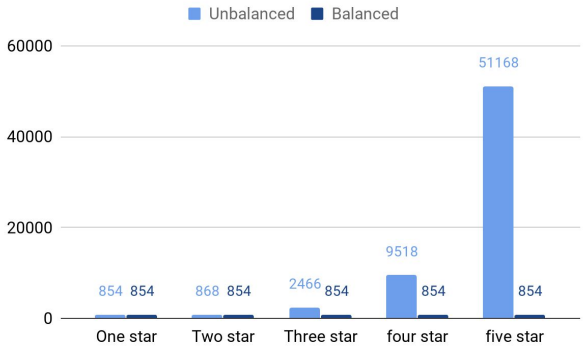

Figure 3. Class distribution of music reviews.

5.2.3 Data normalization

There are two different rating systems in our dataset. The reviews are annotated with a number of stars from one to five; we consider these as an indication that a review describes something very negative (1 star), very positive (5 stars) or in-between (2 3, 4 stars, where 3 is neutral). On the other hand, life memories are rated in a scale from one to seven; we consider these as indications that a life memory describes something very negative (one scale) to something very positive (seven scale). The rating scales are the labels that Pattern and machine classifier will try to predict, in other words the dependent variables. In the experiment that is conducted in this paper we use Pattern as a lexicon based method. As it is explained in section 5.3, Pattern returns a continuous polarity value in between -1 and 1. In order to be able to measure the performance of Pattern, a decision has been made to scale the rating systems in our dataset to an interval in between -1 and 1. By applying rescaling, the difference between the real and predicted value can be correctly measured. In order to achieve rescaling, min-max normalization has been used. To rescale a range between an arbitrary set of values [a, b], the formula is:

$$x' = a + \frac{(x - min(x))\,(b-a)}{max(x) - min(x)}$$

where x is the original value and x' is the normalized value and a,b the min-max values, in our case -1 and 1.

5.3 Pattern

Pattern is a python library, which offers a lexicon-based sentiment analysis with a dictionary of 3198 lemmas (see Figure 1). Pattern's lexicon-based returns a continuous polarity value between -1 and 1, where -1 is very negative, 0 is neutral and 1 very positive. Pattern offers a sentiment function, where you can pass a text as an input and based on the opinion words that are there and the position of the words in a sentence, it will return a polarity result. Pattern's sentiment function takes into account some rules for composition, such as intensifiers and negations. The presence of a word with high intensity, boosts the polarity value of a sentence. For instance, if we pass the following sentence to Pattern: "Het was verschrikkelijk" in english "it was terrible" Pattern will look for opinion words in the sentence and will find one, which is "verschrikkelijk". Afterwards it will return a polarity of -0.9, since in its lexicon the word "verschrikkelijk" has a polarity score of -0.9 (see Figure 4). On the other hand, if we pass the following sentence: "Het was verschrikkelijk goed" in English "it was terribly good", Pattern will find two opinion words, namely "verschrikkelijk" and "goed". In this case Pattern will use the word "verschrikkelijk" as an intensifier and thus it will multiply the intensity score of it, which is 1.9, with the polarity of the word 'goed'. The polarity score result that Pattern will return for that sentence will be 1.9 * polarity(goed), which will return a much more positive polarity score than when the word "goed" is present alone in a sentence. On the other hand, negations in a sentence, such as "The film is not good", are taken into account, in a sense that even in the presence of word "good" which has a positive polarity, Pattern checks for the negation in front of it and returns a negative polarity value instead.

<word form="**goed**" pos="**JJ**" sense="**in orde**" polarity="**0.6**" intensity="**1.0**" />

<word form="**goed**" pos="**JJ**" sense="**correct**" polarity="**0.5**" intensity="**1.0**" />

<word form="**verschrikkelijk**" sense="**eel akelig**" pos="**JJ**" polarity="**-0.9**" intensity="**1.9**" />

Figure.4 Pattern lemma example

5.4 Support Vector Machines

Support Vector Machines (SVM) is a widely used machine learning algorithm. This machine learning algorithm has been applied in many previous research in sentiment analysis and it has shown to give better results not only for sentiment analysis but also for text classification in general compared to other Machine Learning systems, such as Naive Bayes [3,9]. A support vector machine constructs a hyper-plane or set of hyper-planes in a high or indefinite dimensional space, which can be used for regression and classification problems. The basic idea behind the support vector machine is to find a hyperplane that not only separates the feature vectors (see 5.4.3) of each document in one class from the other but also for which the separation is as large as possible. An advantage of SVM is that it can be used for regression and classification problems. In this paper scikit-learn's support vector regressor has been used for training and testing with all parameters set to their default values [1, 12]. The support vector regressor (SVR) is an

extension of the support vector classification (SVC). In the following sections, a discussion is given about whether classification or a regression machine learning system is more suitable for our experiment and the steps that were taken towards implementing the machine learning classifier, which are: tokenization, feature extraction, training the machine classifier and testing it.

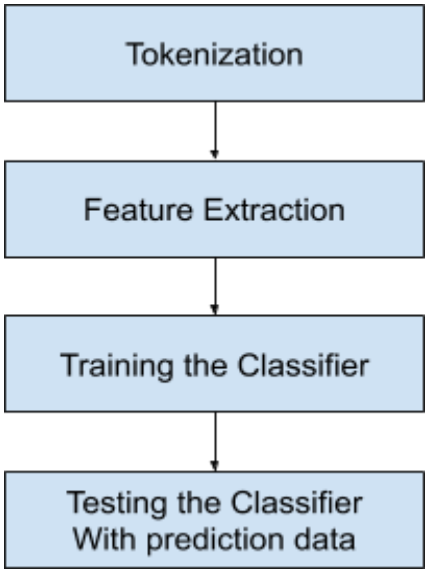

Figure 2. Steps for SVM.

5.4.1 Regression vs Classification

Research in sentiment analysis is mostly focused on predicting whether a document expresses a negative, neutral or positive sentiment and thus deals with categorical classes. In contrast, in this paper the focus is on predicting whether a document is part of one out of five classes for "bol" reviews and one out of seven classes for personal life memories. The dependent variable (the rating scale) is discrete as in a classification task and is ordered as in a regression task. Therefore a decision needed to be made whether the problem should be classified as a regression problem or classification problem. In a regression-based approach the classes are converted into numerical values corresponding to their rank. For instance, a rating of two stars is converted to a numerical value 2.0. On the other hand, in a classification approach the classes are treated as nominal, unrelated classes and the problem is considered as a multi-category classification.

Both approaches have been tested using the SVM classification algorithm and SVM regression respectively, and the difference of the performances of both were negligible. In contrast, research in the past showed that applying regression algorithms to classify documents with respect to an ordinal three-point rating scale provided a higher accuracy than classification algorithms [10]. A decision has been made to use the regression approach because of Pattern. Since Pattern returns a real value in between -1 and 1, it sounds logical to have an SVM classifier that does also the same, since the results of both can be properly compared. In case a classification approach would have been used, then Pattern results had to be classified into categories, by dividing the results into multiple intervals. This would make the whole comparison more complicated and since the difference between the performance of a regression and classification approach for the machine learning system was negligible, the final decision was made to use regression. For our experiment scikit-learn's support vector regressor has been used, which is an extension of the support vector classification algorithm.

5.4.2 Tokenization

Before the feature extraction step takes place, some of the non-relevant words need to be removed from the text documents. This will prevent having a very large feature vector with features that are not relevant for the machine classifier. In order to achieve this, tokenization has been applied. Tokenization is the process of breaking down a sentence into an array of words called tokens, for instance by using white-spaces as token separator and removing or modifying the words. During the tokenization process, non-alphabetical removal, stemming and stop-words removal has been applied. Non-alphabetical characters such as punctuation marks do not contribute to the prediction of sentiments by the machine classifier, therefore those have been removed. In addition, some words express the same sentiments but are written differently when used in different tenses. In order to prevent building a feature vector consisting of words that have the same meaning, but are written differently, stemming has been used. Stemming simply reduces the word to its root word, by removing its suffixes or prefixes. For instance, the word "verschrikkelijk" when stemmed is transformed to the word "verschrik". In this experiment SnowBallStemmer has been used for stemming. SnowBallStemmer is an algorithm provided by Nltk python library and it supports stemming in multiple languages, including Dutch language [4]. On the other hand, stopwords such as "the", "is", "a" etc. do not contribute much in expressing sentiments in a text document, therefore those have been removed. A list of dutch stopwords has been used from the corpus package of Nltk [4].

5.4.3 Feature extraction

A supervised machine learning algorithm cannot use directly the preprocessed and tokenized reviews to train a model, since most of them expect numerical feature vectors with a fixed size rather than an array of words with variable length for each document. In order to address this problem, the following bag-of-features framework has been used. Let $\{f1, \ldots, fm\}$ be a predefined set of m features that can appear in a document; examples include unigrams such as the word 'goed' or the word 'moi'. Let $t_i(d)$ be the significance of feature f_i to the document d, calculated as follows:

$$t_i(d) = TF(f_i, d)\, x\, IDF(f_i)$$

$$IDF(t) = N/DF(t)$$

where $TF(f_i, d)$ is the number of times the term f_i occurs in the document d, and $IDF(f_i)$ is the total number of documents divided by the number of documents in the entire dataset D that contains the feature f_i. Then each document d is represented by the document vector:

$$\overline{d} = (t_1(d), t_2(d), ..., t_m(d))$$

The function $t_i(d)$ presented above is known as the term-frequency inverse document-frequency function and is used for evaluating the significance of a word to a document in a dataset. For implementing the presented bag-of-features framework, the TFI-IDF implementation by Scikit-learn has been used, namely TF-IDF vectorizer [12]. A TF–IDF vectorizer extracts features based on word count, providing less weight to frequent words and more weight to rare words. The parameters used for the vectorizer are: minimum document frequency (min-df) and n-gram range. The min-df parameter will ignore the words that have a document frequency lower

than the given threshold when building the feature vector, in our case the threshold is set to three. The n-gram range parameter specifies the n-gram range to be used when building the feature vector. In our case the parameter is set to the value "1,1" meaning that only unigrams are selected. Unigrams are single words such as "goed" or "moi".

5.4.4 Training and testing data

After the features have been extracted and each document has been transformed to a feature vector, the next step is to split the training and testing dataset. For training the support vector regression classifier the clothing reviews have been selected. The reason why the clothing domain has been selected for training is because the classifier achieved the best performance when trained and tested within this domain. 75% of the clothing dataset has been used for training and 25% of it for testing the classifier. Afterwards the model has also been tested using the whole dataset from the music and personal life memories domains. The same part of data used for testing the SVM classifier was also used to test Pattern.

6. EVALUATION METRICS

As discussed in section 5.4.1, in this paper we use a regression approach to sentiment analysis. Both the support vector regressor and Pattern return a continuous value. In order to evaluate their performance, mean absolute error (MAE) is used. MAE is a common evaluation metric used in regression approaches. One of the advantages of using MAE over other measures such as accuracy, is that it captures the idea that not all answers are the same. If accuracy would have been used, and the problem would be classified as a classification problem, then classifying a document into class five, when in reality the document belongs to class two, it would not be any different from classifying a document into class one. A class one is much nearer to two then class five since the classes are ordinal, but because the accuracy measure treats the data as nominal, it cannot differentiate the distance between classes. On the other hand, MAE captures the exact distance of the predicted value from the real value and therefore it provides much more informative results. Mean absolute error is the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight. MAE is calculated as follows:

$$MAE = 1/n \times \sum_{i=1}^{n} |y_i - x_i|$$

where n is the number of observations, in our case the number of reviews used as testing data, y_i the predicted value and x_i the real value.

7. RESULTS

After testing the SVM classifier and Pattern with testing data across different domains and measuring their performance by using MAE, the results in Table 2 are presented.

Table 2. MAE of SVM and Pattern across domains

Systems	Clothing	Music	Life memories
SVM	0.36	0.48	0.68
Pattern	0.50	0.51	0.65

As it can be seen from Table 2, in the clothing domain SVM has a MAE of 0.36, which is 28% higher than the performance of Pattern in the same domain, which is 0.50. A smaller MAE means a higher performance. In the music domain SVM has a MAE of 0.48, which is around 6% higher than the performance of Pattern in the same domain (0.51) and 25% lower than its own performance in the clothing domain (0.36). Lastly, in the life memory domain SVM has a MAE of 0.68, which is around 6% lower than the performance of Pattern in the same domain (0.65), and around 47% lower than its own performance in the clothing domain.

8. DISCUSSION AND CONCLUSIONS

From the experimental results presented in section 7, we can conclude that a lexicon-based method is not more domain independent than a ML system. In terms of consistency, Pattern seems to produce much better results than SVM. The difference in performance across different domains is smaller compared to SVM's performance across the same domains. In terms of performance, Pattern fails to outperform SVM on music domain, even though the difference is very small. On the other hand Pattern slightly outperforms SVM on life memory domain. Nevertheless, the experimental results are inline with previous work from other studies. As it can be seen in Table 2, SVM clearly outperformed Pattern in the clothing domain. The reason is that SVM has been trained on that domain and this shows that when a machine classifier has a large training annotated dataset, it will outperform a lexicon-based method. Previous research has also shown that a ML classifier outperformed the lexicon-based method when the former is tested within the training domain [17]. Furthermore, research in the past has shown that a ML classifier will not perform very well when used to predict sentiments outside the training domain [2]. This is also inline with our experimental results, since the performance of SVM clearly drops when it is used to predict sentiments from music and life memory domains and the difference in performance with Pattern becomes negligible. It is interesting to see that SVM outperformed Pattern in the music domain, even though its performance is much worse compared to its performance in the clothing domain. One reason could be that the music and clothing domains are somewhat related to each other, even though they were considered as separate domains in our experiment. One could express the same opinions for clothing and music items and therefore the machine classifier was able to predict correctly opinions in the music domain by learning from the clothing domain and in turn outperformed Pattern. On the other hand, the machine classifier performs very badly in the life memory domain and slightly worse than Pattern. A reason could be that sentiments in the life memory domain are expressed totally differently from the sentiments in the clothing domain and therefore the machine classifier finds it hard to classify those. For instance, in the life memory corpus there are memories such as "A friend of mine committed suicide" or "I failed an exam". These sentences express negative sentiments, however for the machine classifier it is hard to predict the polarity of such sentiments when trained in the clothing domain, since such sentences are never expressed in a review of a clothing product. The life memory domain seems to be challenging also for Pattern. It is possible that Pattern's lexicon is missing words such as "suicide" or "fail" and therefore it cannot determine whether those words express a negative or positive opinion.

There are also some limitations with regards to the experiment conducted in this paper. First, the number of domains compared is not enough to make a clear conclusion in the domain independence of Pattern and SVM. Having more testing domains, would show a clearer distinction between the performance of SVM and Pattern across different domains. Second, the negation handling was not considered when

implementing the SVM classifier. Negation handling deals with handling words such as “not good” in a sentence [13]. In our experiment, the SVM will use the words “not” and “good” as separate and not as a single word. Negation handling has shown to increase the performance of SVM in previous work [3].

To sum it up, from the experimental results it is concluded that a lexicon-based method is not more domain independent than a machine learning method. Furthermore, it is concluded that the ML system works as expected within the domain it has been trained and the more we move away from the training domain the lower the performance will be and the difference with Pattern becomes negligible. A “take-home message” that can be derived from the experimental results is the following: In case a ML system is already trained in a specific domain, one can use it to predict sentiment outside the training domain, for instance in domains that lack a large amount or lack a “good-quality” of labelled data. The system will not do very great outside the training domain but also not too bad compared to a lexicon-based system. On the other hand, if a ML system needs to be trained from scratch and there is not enough labelled data for the domain you want to predict sentiments on, a better choice is Pattern. The reason is that it is faster and easier to use and compared to the ML system the results with regards to performance are the same.

9. REFERENCES

[1] Alex, J, S., Bernhard, S. 2004. A Tutorial on Support Vector Regression - Statistics and Computing archive Volume 14 Issue 3, August 2004, p. 199-222.

[2] Aue, A., and Gamon, M. (Sept. 2003). Customizing sentiment classifiers to new domains: A case study. In Proceedings of recent advances in natural language processing (RANLP) (Vol. 1, No. 3.1, pp. 2-1).

[3] Bo, P., Lillian, L., and Sh., Vaithyanathan. 2002. Thumbs up? *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - EMNLP '02* (2002). DOI=http://dx.doi.org/10.3115/1118693.1118704

[4] Bird, S., Edward, L and Ewan, K. (2009), *Natural Language Processing with Python*. O’Reilly Media Inc.

[5] Choy, Y. and Lee., H. 2017. Data properties and the performance of sentiment classification for electronic commerce applications. *Information Systems Frontiers* 19, 5: 993–1012. DOI= http://doi.org/10.1007/s10796-017-9741-7

[6] Das., S. and Chen., M. 2001. Yahoo! for Amazon: Extracting market sentiment from stock message boards. APFA’01.

[7] Hu, M., and Bing, L. "Mining and summarizing customer reviews." *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2004.

[8] Jie, T. and Xing, F. 2020. Toward multi-label sentiment analysis: a transfer learning based approach. *Journal*

[9] *of Big Data* 7, 1. DOI= http://doi.org/10.1186/s40537-019-0278-0

[10] Joachims, T. (1998, April). Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning* (pp. 137-142). Springer, Berlin, Heidelberg.

[11] Koppel, M and Jonathan, S. 2005. The importance of neutral examples for learning sentiment. In Workshop on the Analysis of Informal and Formal Information Exchange during Negotiations(FINEXIN).

[12] Mukhtar, N., Khan, M. A., & Chiragh, N. (2018). Lexicon-based approach outperforms Supervised Machine Learning approach for Urdu Sentiment Analysis in multiple domains. *Telematics and Informatics*, *35*(8), 2173-2183.

[13] Pedregosa *et al.* 2011. Scikit-learn: Machine Learning in Python, , JMLR 12, pp. 2825-2830, 2011

[14] Sanjiv, D. and Mike, C. 2001. Yahoo! forAmazon: Extracting market sentiment from stock message boards. InProc. of the 8th Asia PacificFinance Association Annual Conference (APFA2001).

[15] Smedt, G. and Daelemans, W. 2012. Pattern for Python. University of Antwerp, Belgium.

[16] Xing, F. and J., Zhan. 2015. Sentiment analysis using product review data. *Journal of Big Data* 2, 1. DOI = http://doi.org/10.1186/s40537-015-0015-2

[17] Xiaowen, D., Bing, L., and Philip S. Yu. 2008. A holistic lexicon-based approach to opinion mining. In Proceedings of the 2008 International Conference on Web Search and Data Mining (WSDM ’08). Association for Computing Machinery, New York, NY, USA, 231–240. DOI:https://doi.org/10.1145/1341531.1341561

[18] Zhang, H., Gan, W., and Jiang B. 2014. Machine Learning and Lexicon Based Methods for Sentiment Classification: A Survey. 2014 11th Web Information System and Application Conference (2014).