Universität Zürich[UZH]

Bachelor's Thesis
to obtain the academic degree
Bachelor of Arts
at the Faculty of Arts and Social Sciences

Quantifying the Polarity of Noun Phrases

Author: Berna Ilke Ersoy
Matriculation Number: 19-748-979

Supervisor: Dr. Manfred Klenner

Department of Computational Linguistics

Submission Date: 01.06.2022

Abstract

While sentiment analysis is a popular NLP subfield, not every area within gets the same amount of attention. For example, classifying the sentiment of texts seems to be quite in demand, however, rarely does it also include determining how positive or negative they are, especially when it comes to phrase level. To tackle that problem, this thesis deals with the quantification of polarities of noun phrases, i.e. determining how positive or negative a noun phrase is and calculating a score. This enables the creation of a polarity ranking for all phrases, and as such a task can be challenging and time-consuming for humans, this thesis aims to provide approaches that automatize this.

Zusammenfassung

Obwohl die Sentimentanalyse ein beliebtes NLP-Teilgebiet ist, wird nicht jedem Teilbereich innerhalb die gleiche Menge an Aufmerksamkeit geschenkt. Zum Beispiel scheint die Klassifizierung des Sentiments von Texten sehr gefragt zu sein, aber nur selten wird dabei auch festgehalten, wie positiv oder negativ sie tatsächlich sind, vor allem, wenn es sich um die Phrasenebene handelt. Um dieses Problem anzugehen, beschäftigt sich diese Arbeit mit der Quantifizierung der Polaritäten von Substantivphrasen, d.h. der Bestimmung, wie positiv oder negativ eine Substantivphrase ist, ausgedrückt in Zahlen. Dies ermöglicht die Erstellung eines Polaritätsrankings für alle Phrasen. Da solch eine Aufgabe für den Menschen schwierig und zeitaufwendig sein kann, ist das Ziel dieser Arbeit, Ansätze zu liefern, die dies automatisieren.

Acknowledgement

Firstly, I would like to thank my supervisor Manfred Klenner for inspiring me to find an idea for my thesis and also for supporting me by helping with any questions and problems I had. Furthermore, I want to thank Dario Mulé for assisting me with the annotations and keeping my motivation up all this time. Last but not least, I also want to thank my family for supporting me at all times.

Contents

List of Figures

List of Tables

List of Acronyms

BWS	Best-Worst Scaling
kNN	k-Nearest-Neighbors
MSE	Mean Squared Error
MLP	Multilayer Perceptron
NLP	Natural Language Processing
NP	Noun Phrase
POS	Part-Of-Speech
SVR	Support Vector Regression

1 Introduction

1.1 Motivation

Sentiment analysis has gained quite the popularity along with the rise of social media; and while it continues to do so, we continue to get access to new data and possible research areas. There are many interesting ones to explore (which I will also talk about in section 2.1), but one I find particularly intriguing is polarity analysis. When the polarity is mentioned in the field of sentiment analysis, usually one refers to labeling things as 'positive', 'neutral', or 'negative'. This thesis will try to do exactly the latter to noun phrases (phrases with a noun as their head, e.g. 'the huge beautiful tree'). To add a little twist to the task, it will not simply be a matter of assigning labels to noun phrases, but also of quantifying how 'positive' or 'negative' they are while also creating a ranking. This will be done by trying out different methods and evaluating their output's quality.

1.2 Research Questions

The goal of this paper is to find a way to quantify the polarity of noun phrases with machine learning. This essentially means that each noun phrase should get a score and be ranked, here in this paper, from highest to lowest score, which should correspond from most positive to most negative.

As there is no gold standard for this specific noun phrase task, the main goal will be to find a suitable approach to the problem, which will include training on already existing related material, or trying to generate an own silver standard to train on by trying out different techniques to tweak the scores such that it can rank noun phrases as well as possible. Another task will be to find appropriate noun phrases, allowing us to generate rankings in the first place. Finally, all approaches will be compared manually and with evaluation metrics. Tasks that require some type of programming will be done with the programming language Python 3. All programs

and resources that were created or used can be found on GitHub[1].

1.3 Thesis Structure

This chapter introduced the motivations and the main task of this project. Chapter 2 will provide knowledge that is necessary to be able to get a good understanding of later chapters and talk about related work. In Chapter 3, we will take a look at what data, methods, and models are used to solve the aforementioned tasks. Chapter 4 is where the results of the different approaches will be shown and evaluated. Chapter 5 will be used to wrap everything up and to provide a final impression.

[1] https://github.com/bernai/Quantifying-NP-Polarity/

2 Background

2.1 Sentiment Analysis

Sentiment Analysis, also used synonymously with Opinion Mining, is an area concerned with investigating everything that has to do with opinions, feelings or attitudes (Liu, 2012). While this investigation can be done with statistical methods, machine learning, or natural language processing (NLP), the focus of this thesis lies mainly in machine learning methods, while additionally using a lexicon-based approach to generate data the machine learning systems can work with.

There are many application areas, like the automatic extraction of emotions from books, articles or social media content, classification tasks, which for instance involve finding out whether a post can be considered funny or not, or measuring polarities (positive/negative) that can also involve regression tasks to calculate intensities (e.g. how positive/negative something is).
The latter represents the goal of this thesis, as 'Quantifying the Polarity of Noun Phrases' in less fancy terms is essentially analyzing how positive or negative noun phrases are.

It is also useful to realize that sentiment analysis can take place on different levels. Liu (2012) gives a good insight into the whole thing. Document level analysis, as the name suggests, deals with the sentiment of a whole document. An example can be finding the emotion a document expresses overall. There is also sentence level analysis, where only a single sentence is the main concern, for example when detecting irony in a single sentence. Finally, there is the entity and aspect level analysis, which is more detailed and can be used to find out what the attitude or emotion towards a certain entity, e.g. a specific phone model, is.
If one were to classify what level is dealt with in this thesis, we could say that the noun phrase level is closest to sentence level analysis, as the entity level analysis involves a more detailed approach than only quantifying polarities of noun phrases.

2.2 Related Work

2.2.1 Valence Shifters

When trying to find out whether a sentence is positive or negative, one might think that adding up positive and negative words to yield a result could be good enough. Polanyi and Zaenen (2006) elaborate on this and show why this is not the case, even though it was a common method at the time the paper was written, by showing factors that influence the classification of polarity in texts.

Firstly, they illustrate the observation that texts can contain similar information, but have different polarities. In one of their examples, they changed 'the eighteen year old', which is a rather neutral expression, to 'the young man', creating a more positive version of the same fact. Cases such as these have led to the differentiation between negative and positive words becoming important. However, as mentioned, these polarities can be changed by several interactions, which are called valence shifters, as they change the valence/polarity of words and phrases. These shifters include negatives, modals, intensifiers (e.g. 'very'), irony and many other categories. For example, it makes a difference whether something 'should be good', 'is good' or 'is not good', which shows that many things need to be considered at once when analyzing polarities.

Also, depending on what is analyzed, it may make sense not to count the word towards the polarity at all. If we are interested in a review of a specific entity like a company's product, and within the review the competitor's products are mentioned along with some positive words about their product, one cannot just attribute this positively towards the sentiment of the reviewed product. When Polanyi and Zaenen (2006) wrote their paper, this type of entity analysis had serious limitations. By now, progress has been made in this field of sentiment analysis, where entity level is even considered a separate depth-level, as we have seen in the previous section.

2.2.2 Sentiment Analysis with Lexicons

Taboada et al. (2011) take the valence shifters matter further and create the reliable SO-CAL (Semantic Orientation CALculator) system that takes them into account to calculate semantic orientation. This orientation value includes word polarity and intensity and relies on different lexicons, which they created manually. The lexicons contain words separated by type, e.g. intensifiers, nouns, verbs, and contain their prior polarity, which is the internal polarity of the word without taking context into

account. Combining prior polarity and valence shifter information within a text with a series of calculations, leads to a final semantic orientation value of texts, the polarity either being 'positive', 'neutral' or 'negative', and the intensity value being a floating-point number that represents how strong the polarity label is.

When creating the lexicon for the SO-CAL, they eliminated ambiguous words that mostly did not add to the sentiment of a text, and also refrained from using too many words to avoid further unnecessary confusion dealing with ambiguities. The basis of the quality evaluations of their lexicon were manual, human annotations. The SO-CAL system was also tested on other lexicons, among others one by Google, but seemed to perform best on the one they created manually. Taboada et al. (2011) assumed this to be due to the minimization of ambiguity noise in their approach. Additionally, it was established that previous research concluded that machine-learning methods perform better when evaluated on one single domain, but lexicon-based systems can catch up or even surpass them in cross-domain settings, which is why the SO-CAL system might turn out to be a more useful tool for cross-domain sentiment analysis.

The silver standard generation in this thesis uses some processes mentioned in this paper, as it uses lexicons to calculate a polarity score and takes some valence shifters into account. Even though the focus will be mainly on machine-learning approaches, the silver standard alone will also be evaluated regarding its ranking quality in Section 4.2.1.

2.2.3 Sentiment Analysis with Machine Learning

One major paper dealing with the topic of this thesis, and the one I will keep coming back to, is by Kiritchenko et al. (2016). Here, they cover the 7th task of the semantic analysis workshop SemEval in 2016, where the main goal was to rank single words or phrases in three specific domains according to their polarity. The three included one set containing general English words and phrases, one set with English tweets, and one with Arabic words and phrases. As there was no official training data given, a major challenge was to find suitable resources for the task. In the end, the results of the contestants were evaluated by a manually created ranking that contained all examples from the respective sets. These manual rankings were created by a neat method called Best-Worst Scaling (BWS), including the General English Words and Phrases lexicon (SCL-NMA) that we will use later on in Section 4.3. This method works by letting a group of people determine which the best and which one the worst term is in the sense of the task. Here, best or worst referred to most positive

or most negative, and the annotators had to look at 4 terms for each best or worst selection.

With the help of all this data, they were able to calculate a score for every word and phrase in the set, and rank everything. This so-called gold standard was eventually used for evaluation purposes. Gold standards are datasets that are created by manual human annotation as in Kiritchenko et al. (2016), while datasets created with the help of automation are called silver standards. Usually, gold standards are higher in quality, but require quite the resources to do so as it is apparent here.

In the end, there were three submissions for each task and the best approach was found by comparing the gold standard ranking to the ranking of each team. This analysis was done by a rank coefficient called Kendall's Tau (Kendall's τ), which was also the official metric for evaluation. Kendall's Tau measures how related two rankings are, which is exactly what was needed here to find the best team. Spearman's Rho (Spearman's ρ) is another rank correlation metric and was used alongside Kendall's Tau as an added metric.

I will talk about these evaluation metrics in more detail in Section 3.8, as the same metrics will be used to compare how the models of this thesis compare to the models of the contestants later on in Section 4.3.

Another paper, which deals specifically with quantifying polarity, was written by Göhring and Klenner (2022). Their goal was to build a system that reliably calculates the polarity of German noun phrases while also finding actors that are mentioned. The detection of actors was solved with a binary classification approach, which means that for each word, the model calculated whether a word was an actor or not. In order to calculate polarity, they created a silver standard, as there were no gold standard datasets available for German, and trained regression models with its help. The generation of the set involved the usage of a polarity lexicon that contained the strength of a word, the conveyed base emotion if there is any, a behavioral tag defining whether it is passive or active and its appraisal category (Göhring and Klenner, 2022). The defined base emotion includes one of the eight emotions defined by Plutchik (1980), namely anticipation, fear, anger, surprise, joy, disgust, sadness, or trust. The appraisal category builds on the appraisal theory, a linguistic theory by Martin and White (2005), that is utilized to analyze the sentiment in English text or speech. Appraisal has three categories, namely gratitude, dealing with changing the strength or focus of the opinion voiced, engagement, which deals with the positions of the speakers or authors regarding statements, and attitude, which is about whether a positive or negative sentiment is expressed. The lexicon

uses appraisal in the sense of attitude, which can be divided further into whether a word is an emotion (e.g. 'glad'), judgmental (e.g. 'merciless') or factual (also called appreciation, e.g. 'legitimacy'), and was also used in the lexicon in this manner.

After the regression models were trained using the silver standard and word embeddings (see Section 3.7 for an explanation), the best one was chosen and used for the quantification of noun phrases. Both this and the actor classification model were evaluated on newly generated noun phrases that were unknown to the model. In total, 670 NPs were manually evaluated by two annotators by looking at 5 NPs each time and determining outliers in the ranking. In the end, the evaluation resulted in 83.15% of the NPs placed correctly on average within the ranking.

This evaluation technique will also be used in this thesis to determine the quality of the models when evaluating noun phrases.

2.2.4 Sentiment Analysis using Multi-Task Learning

Another interesting approach to sentiment analysis is realized by Tian et al. (2018), where instead of purely using regression or classification systems for polarity analysis, they inspect the effect of multi-task learning on performance. This works by letting the models' tasks consist of predicting sentiment scores, and classifying polarity (positive/negative) and/or intensity (strong, medium, weak, neutral) at once, while treating polarity and intensity classification as secondary tasks.

They analyzed each combination of sentiment score regression with polarity and intensity classification on unimodal and multimodal approaches, where visual, vocal or verbal information were the modalities. Ultimately, the results suggested that multimodal (using multiple modalities) approaches generally did not benefit much from multi-task learning, while all unimodal approaches had improved their performance with multi-task learning.

I will not use multi-task learning for the tasks of this thesis, however, seeing the results of Tian et al. (2018), it would be an interesting approach for future work.

3 Data and Methods

3.1 Overview of Methods

Table 1 shows the different approaches that have been implemented for the polarity quantification task. The first approach involves the generation of an own silver standard by using different lexicons, which also directly leads us to the second method of using the silver standard scores along with pre-trained word embeddings to train different machine learning models. The last approach uses a valency, arousal, and dominance lexicon as a base for training, also using pre-trained embeddings. This Chapter serves to discuss the three techniques in detail.

Approach	**Lexicons used**	**Resources Used**
Silver Standard only	NRC Emotion Lexicon, Subjectivity Lexicon MPQA, PoLex (english polarity lexicon)	None
Silver Standard as Training Data	NRC Emotion Lexicon, Subjectivity Lexicon MPQA, PoLex (english polarity lexicon)	pre-trained fasttext/ word2vec embeddings
NRC-VAD as Training Data	NRC-VAD	pre-trained fasttext/ word2vec embeddings

Table 1: Different Approaches for Polar Quantification

3.2 Extraction of Noun Phrases

To analyze noun phrases at all, one needs some first. With the help of the dataset provided by Greene and Cunningham (2006) and some Python modules, many noun phrases could be extracted. For this, I used the summaries of BBC news articles with the label 'entertainment', which included a total of 386 articles. The sentences were tokenized with the Python module NLTK and the noun chunks (base noun phrases) were extracted with spaCy. This led to 35'000 noun chunks that had to be cleaned in order to be useful, as there were many repetitions and noun phrases consisting

only of a single pronoun. As this would lead us to a rather monotonous ranking, noun phrases that already occurred or were only a single word were removed. In the end, almost 8'000 noun phrases, for example, 'an extremely professional and competent broadcaster', were ready to be processed by the models.

3.3 Lexicons

This section contains all lexicons that were used for the project together with a short description for a better understanding.

3.3.1 Polarity Lexicon

The polarity lexicon from the PolArt project by Klenner et al. (2009) contains English words along with their appraisal category, degree modifier category (whether it is a shifter, diminisher, or intensifier, e.g. 'very' is an intensifier) and its POS tag. This kind of information helped with the generation of a silver standard.

3.3.2 EmoLex

Mohammad and Turney (2013) created an NRC Word-Emotion Association Lexicon (EmoLex) using the BWS method (see Section 2.2), and contains words together with the base emotion they convey. Additionally, it shows the sentiment of each word (positive/negative). In total, it contains 14'182 words. This lexicon was used for the generation of the silver standard as well.

3.3.3 MPQA Subjectivity Lexicon

The MPQA Subjectivity Lexicon by Wilson et al. (2005) contains many features of words, as for example the polarity of the word when there is no context (prior polarity), the POS tag or also whether the word is strongly subjective or not. As with the previous lexicons, I used a part of the data for generating the silver standard.

3.3.4 NRC-VAD

The NRC Valence, Arousal, and Dominance Lexicon (NRC-VAD) by Mohammad (2018) is another crowdsourced lexicon that needed the BWS method for its creation.

Each of the around 20'000 words get a valency, arousal and dominance score ranging from 0 to 1. Valency determines how positive or negative the word is, while arousal shows the active or passiveness, and dominance shows its strength. This lexicon was used as a gold standard to train the models in Section 3.9.2.

3.3.5 SCL-NMA

Section 2.2 already mentioned this lexicon, which as a reminder, was created with the BWS method and includes general words and phrases in English with their polarity score. It is also referred to as the General English Sentiment Modifiers lexicon. The newest version contains 3207 terms, with values ranging from -1 to 1 for each term. This thesis uses this lexicon for two main reasons. Firstly, to compute a silver standard dataset by using its words and phrases, and secondly to compare the trained silver and gold standard models to the results in Kiritchenko et al. (2016).

3.4 Programming

As previously mentioned, the programming language used for this project is Python 3. With many useful packages available, it makes programming as fast and as simple as possible. The main libraries used for this project were spaCy, scikit-learn, and NLTK. SpaCy (Honnibal and Montani, 2017) is a package that can process natural language, and in our case, as briefly mentioned before, useful for extracting noun chunks from the previously mentioned dataset. NLTK (Bird et al., 2009) is another package for natural language processing, which was used to lemmatize words in phrases when the original word could not be found in the aforementioned lexicons. Last but not least, scikit-learn (Pedregosa et al., 2011) is where the machine learning models come from. It is an extremely practical package that makes life easier for solving machine learning tasks. As quite a few different models were used in this thesis, Section 3.6 will deal with their basics.

3.5 Machine Learning and Lexicon-Based Approaches

Both machine learning and lexicon-based approaches are used in this thesis. While lexicon-based approaches require no training data, they still require a lexicon as a foundation as its name suggests. If the lexicon is not large enough, this can lead to

missing information during processing. As it is not possible to cover every possible future case that might occur in a lexicon, this is bound to happen sooner or later. In Section 3.9.1, we will see what this problem can look like when generating a silver standard with the help of several lexicons. The main problem of the lexicon-based approach can be solved with machine learning-based systems, as also cases are covered that were not previously seen during training. The difficulty in such an approach lies in finding suitable training data, as firstly, there needs to be enough data to train on, and secondly, the information must be of reasonable quality.

3.6 Models

All models in this project use supervised learning methods, which essentially means that the models learn with pre-labeled examples. The goal is that the models can provide an output for a given input, in our case, the scores for certain words when given word embeddings, which then are used to rank phrases properly.

In total, four different regression models will be compared to each other, namely Linear Regression, MLP, kNN, and SVR. Regression is an analysis method where a relationship between a dependent and independent variable is calculated. In our case, the noun phrases would be the independent variable, and the polarity scores the dependent variables. In the following subsections, each model will be described in detail.

3.6.1 Linear Regression

Linear regression is a model that predicts outputs by setting up a linear function that maps from x to y with the least amount of error possible. As mentioned by Kumari and Yadav (2018), linear regression can also formally be described by the search for a function $y = mx + c$, where x is the input and the independent variable, and y is the prediction and dependent variable.

3.6.2 MLP Regressor

The Multilayer Perceptron Regressor is a simple neural network model. Figure 1 is an illustration taken from the scikit-learn (Pedregosa et al., 2011) documentation and shows a Multilayer Perceptron, which is used for classification tasks but works similarly as the MLP Regressor.

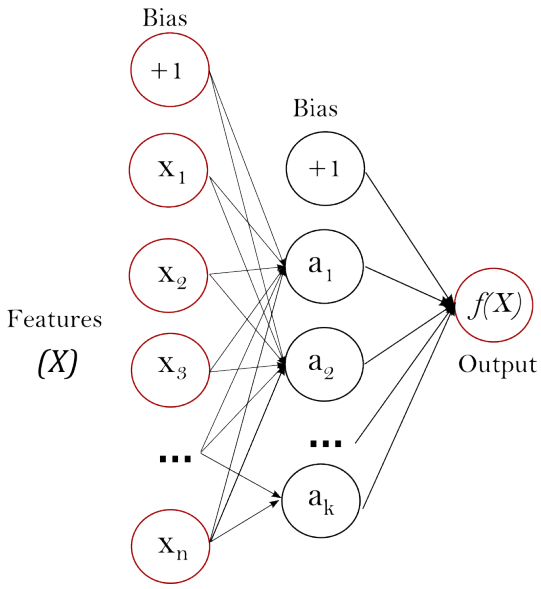

Figure 1: MLP illustration by scikit-learn (Pedregosa et al., 2011)

The MLP network starts by taking in a number of features, represented by the red set of nodes on the left side. These values are altered by the nodes in the middle, which is called the hidden layer, by a series of calculations whose explanations I will skip for the sake of simplicity. In this figure, there is one hidden layer, but it is possible to have more than one. Eventually, the last layer calculates all inputs from the hidden layer into a value. To convert this value to an output label in a classification task, there is a need for an activation function, which leads us to the difference between the MLP classification and regression model. The regressor does not need a label as output, hence there is no need for an activation function, which is the difference between both model types (Pedregosa et al., 2011).

3.6.3 kNN

When trying to predict an output, Nearest Neighbor algorithms try to find the closest values to a provided input. These so-called neighbors that form the basis of the prediction are found among examples that were seen during training. The thought here is that similar inputs should have similar outputs (Shalev-Shwartz and Ben-David, 2014). In this paper, the focus lies on regression, even though the algorithm can be used for classification problems as well.

As the name of the k-Nearest-Neighbors (kNN) algorithm suggests, there is a main

variable k, that describes how many neighbors we want to retrieve at a time. If for example $k = 10$, the ten closest points are calculated and averaged to predict a value. In the Python module scikit-learn (Pedregosa et al., 2011), by default, the kNN regressor calculates the mean of all k neighbors with equal weighting.

3.6.4 SVR

The Support Vector Regression (SVR) algorithm, in contrast to e.g. linear regression, does not look for a single line that fits best, but searches for a plane with the addition of a margin that gives the values a certain range they can be in (Awad and Khanna, 2015).

In Figure 2, there is a 2D illustration of an SVR function. The line $\mathbf{w}^\top \mathbf{x}$ and two adjacent lines that are ϵ away. The goal is to minimize the loss by keeping the predicted values within this ϵ boundary, as the points only count towards the loss if they are on the epsilon lines or further away. These outliers are marked on the figure with a red border and are called support vectors.

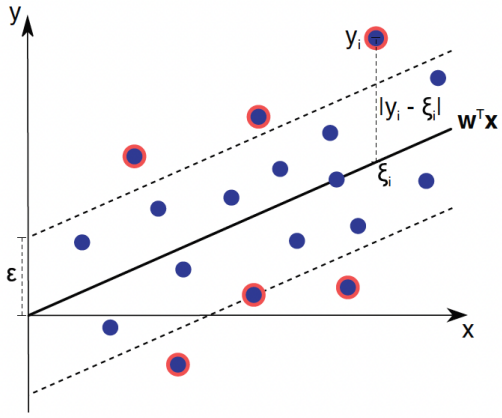

Figure 2: SVR function illustration by Rosenbaum et al. (2013)

Another fact to consider with SVR is that there are a few kernels one can choose from. There are a few kernels one can use for the calculation of the decision boundary and line, among others the kernels 'linear' and 'RBF'. The linear kernel calculates a linear function for the hyperplane, whereas the RBF kernel creates a non-linear one. As the RBF kernel deals with more complex calculations, the training time is rather long compared to models with linear kernels, which is problematic for larger datasets.

Nevertheless, RBF kernels usually perform better. Keerthi and Lin (2003) even concluded that for Support Vector Machines (the classifier version of SVR) there

is no need to even consider linear kernels, as the RBF with the right parameters always performs equally or better.
There is only a humble amount of training data for the task of this thesis, thus, the RBF kernel was used for better results.

3.7 Word Embeddings

Word embeddings are representations of words and capture their meaning in a way that can be understood by computer systems (Srinivasan, 2018). An assumption is that related words appear in similar environments (Srinivasan, 2018), and the goal of word embeddings is to capture that. One way is to map them onto a vector space, which is the case for the embeddings that are used for this thesis. Here, the idea is that words appearing in similar contexts have similar vectors. This can be of use when utilizing the models by representing noun phrases as vectors, so similar noun phrases can get similar results. I will use fasttext and word2vec word embeddings to do exactly that.

The first pre-trained representation (crawl-300d-2M-subword), created by Mikolov et al. (2018), was trained using fasttext on Common Crawl and contains 2 million word vectors with subword information (character n-grams). The advantage of this type of representation is that words that were not seen during training can still get vectors, as fasttext can use subword information and get embeddings for words split into n-grams by the utilization of the character n-gram vectors (Mikolov et al., 2018).

The second pre-trained representation (GoogleNews-vectors-negative300) contains 3 million word vectors and was created by using word2vec (Mikolov et al., 2013). In contrast to the fasttext embedding, words that do not exist in the trained model do not get a vector, which is why in those cases the words have to be represented as a vector filled with zeros.

3.8 Evaluation Metrics

To evaluate the models, the R^2 score and the MSE will be utilized and calculated in scikit-learn.

The R^2 score, also called the coefficient of determination, shows the variance between the values of a dataset and the predictions of a model (Kharwal, 2021). It can be

used for both the test and training data. If calculated on the training set, one can see how well the model fits the data. On the test set of the model, it shows how well the model is able to predict new values, as a higher score shows a high correlation between the inputs and predicted values. There is no minimum score, but the maximum value is 1, which is the best possible result.

The mean squared error (MSE) can be used to measure the error of a model. It is calculated by taking the difference between the actual and predicted value, and then squaring the difference. The advantage of squaring the values is that predictions that differ strongly from the actual value are penalized more severely. This also means that the closer the MSE is to 0, the better the model is in predicting without error.

To compare rankings, another way of calculating the quality of the model outputs is needed. As indicated earlier in Section 2.2, Kendall's Tau and Spearman's Rho are the perfect candidates for this purpose.

Kendall's Tau is a rank correlation metric with which one can compare two rankings (Kendall, 1938). With its help, the models of this thesis can be compared to the results in Kiritchenko et al. (2016) later in Section 4.3. For the calculation, scikit-learn will be used, where Kendall's Tau is calculated as follows according to its documentation:

```
tau = (P - Q) / sqrt((P + Q + T) * (P + Q + U))
```

The variable P denotes the number of concordant pairs, which is the amount of pairs that are the same for both rankings, and the variable Q denotes number of discordant pairs which is the exact opposite. T and U are variables that are used for rankings with ties, which will not be needed for the cases in this thesis. The result can range from -1 to 1, where 1 is a perfect correlation, and -1 is no correlation at all.

Another correlation rank is Spearman's Rho (Spearman, 2010), which will be used alongside Kendall's Tau to compare my results to Kiritchenko et al. (2016). Again, the metric will be calculated by scikit-learn and ranges from -1 to 1.

3.9 Methods in Detail

3.9.1 Silver Standard

The silver standard generation required three different lexicons, which were converted into Python dictionaries. The contents of the dictionaries and important steps of the generation can be seen in Figure 3. The process for getting scores begins by splitting an input phrase into words. Each word is then looked up in the EmoLex, and if it exists, a base score is calculated by counting together the conveyed base emotions. The word is assigned a score based on how many positive or negative words it represents, which automatically leads to a score of 0 if the word is not in the EmoLex. This base score is tweaked by the strength of the word, which can be extracted from the Subjectivity Lexicon MPQA. Strong words get an intensified score, weak words keep their score. Then, based on what additional attributes the PolArt lexicon assigns to the word, the score is altered again until the final score is reached. After each word in the phrase has obtained a score, they are averaged, resulting in the polarity score of the phrase after being squeezed into a number between -1 and 1.

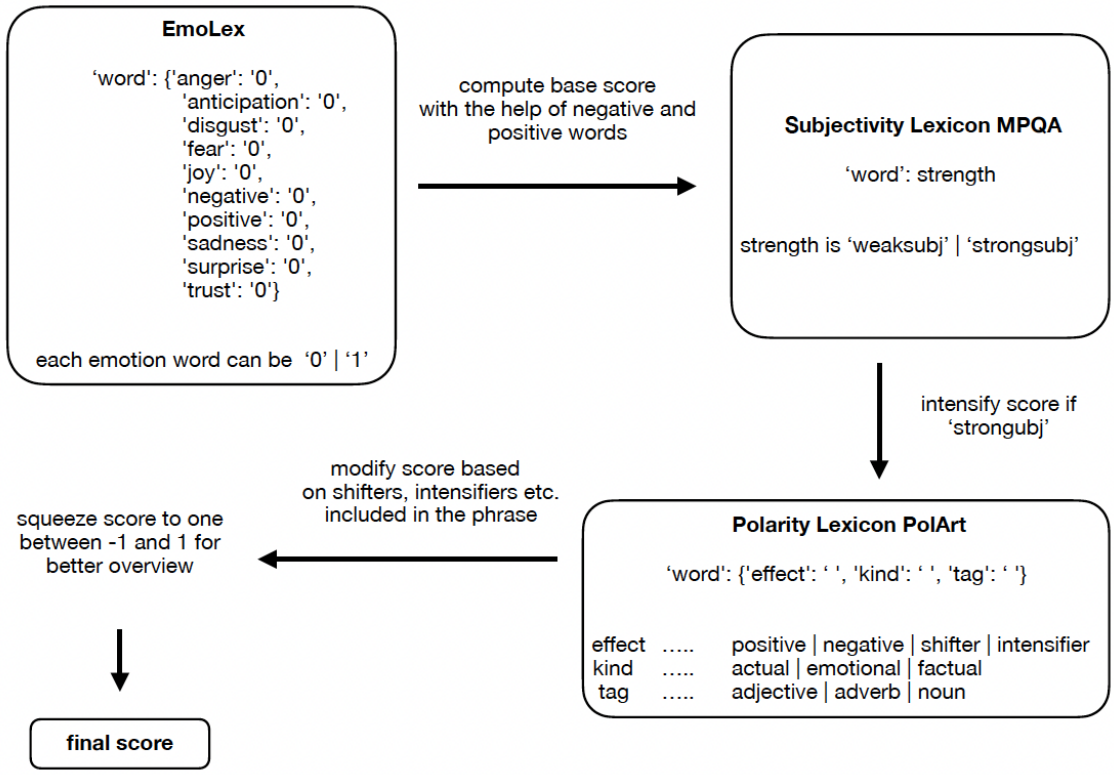

Figure 3: Generating Score for Silver Standard

In order to create a silver standard dataset that the models can use for training, the phrases from the SCL-NMA lexicon were used. This also allows us to evaluate later on, whether the silver standard can keep up with the gold standard, and if training with embeddings leads to any significant improvements.

In Table 2, there are score calculation examples on a few phrases from the SCL-NMA lexicon. The scores can range from 1 to -1, where 1 is most positive and -1 is most negative. It is apparent that the results are not quite perfect, for example, words that do not exist in the EmoLex get no base score, and some words have such a negative score that it affects the whole phrase very strongly. Also, at first glance, one might think that the negation 'not' was ignored in the last case. However, I decided to treat negations as score diminishers rather than score shifters, as usually taking the opposite score is not the better answer, like in this case. The diminishing effect seems to have been smaller than the negative score of the word 'bad', which must have led to this score.

phrase	**phrase score (rounded)**
very very happy	0.999
was so pleased	0.978
must be better	0.421
no support	0.291
must accept	0.0
quite understand	0.0
not too bad	-0.999

Table 2: Examples: Silver Standard Generation SCL-NMA

Now that a silver standard was available, the training could begin. For training, the data was split into 90% training data and 10% test data. This permits calculating the performance metric for both sets and seeing if the models are of any use. As for the input, phrases were turned into word embeddings by taking the average embedding of all words in the phrase. Using word embeddings allowed the model to predict scores after training more reliably, as there was some internal representation of words, even if they were never seen before.

Table 3 shows the results of the evaluation metrics for each model. If one were to judge according to the metrics, there seem to be a few promising models, as fasttext MLP. However, after some manual evaluation, I realized that the metrics did not correlate with the quality of the ranking at all.

w2vec		linear	MLP	kNN	SVR
MSE	**training**	0.272	0.207	0.355	0.225
	test	0.303	0.303	0.396	0.293
R^2	**training**	0.400	0.544	0.217	0.503
	test	0.367	0.367	0.173	0.388

fasttext		linear	MLP	kNN	SVR
MSE	**training**	0.235	0.007	0.140	0.116
	test	0.300	0.189	0.230	0.286
R^2	**training**	0.483	0.984	0.691	0.744
	test	0.371	0.603	0.510	0.402

Table 3: Metrics: word2vec and fasttext embedding models

Table 4 shows the same phrases as in Table 2, but this time ranked according to the scores calculated by the word2vec and fasttext SVR models, as well as the MLP fasttext model. Metric-wise, the SVR word2vec model should perform the worst. Even so, upon looking at the scores and rankings, it is clear that the MLP model makes the least sense while the SVR word2vec model creates the most reasonable ranking. A possible explanation for why the metrics do not correlate with the final performance on general phrases could be the fact that the silver standard itself has difficulties with the phrases of this lexicon, and a low error and a good coefficient of determination only show that the model is close to the predictions of the silver standard. Thus, at least for the silver standard, comparing models by looking at the metrics is not the best option. After looking at all the outputs manually, the SVR word2vec model seemed to still be the best overall among all trained models.

phrase	SVR w2vec
very very happy	0.769
was so pleased	0.535
no support	0.312
must be better	0.207
quite understand	0.165
must accept	-0.014
not too bad	-0.488

phrase	SVR fasttext
must accept	0.551
must be better	0.446
very very happy	0.4292
no support	0.236
quite understand	0.176
was so pleased	0.0369
not too bad	-0.038

phrase	MLP fasttext
quite understand	0.251
must accept	0.149
very very happy	0.128
was so pleased	-0.063
not too bad	-0.199
no support	-0.279
must be better	-0.290

Table 4: Silver Standard: Examples for Model Prediction on SCL-NMA

3.9.2 Gold Standard (NRC-VAD)

As previously mentioned, there is no gold standard that focuses on noun phrases. This is why I wanted to test out, whether training models to predict single word polarity with a word embedding as an input would lead to a good performance for this task. The NRC-VAD was the perfect candidate to fit into this role, as there were a few scores to choose from. After trying combinations of valency, dominance and arousal, I figured that using valency alone led to better models, and discarded arousal and dominance. As with the silver standard models, the input was in the form of averaged word embeddings, and the dataset was split into 90% training data and 10% test data.

word2vec		linear	MLP	kNN	SVR
MSE	**training**	0.020	0.003	0.021	0.007
	test	0.020	0.030	0.027	0.016
R^2	**training**	0.573	0.936	0.556	0.843
	test	0.573	0.385	0.446	0.658

fasttext		linear	MLP	kNN	SVR
MSE	**training**	0.020	0.004	0.015	0.190
	test	0.022	0.018	0.018	0.021
R^2	**training**	0.563	0.901	0.670	0.588
	test	0.544	0.622	0.601	0.600

Table 5: Metrics: word2vec and fasttext embedding models

Now that the basis is a gold standard, the metrics have more meaning than with the silver standard models. Two models were superior to the others, and are marked in Table 5. The SVR model was trained with word2vec embeddings and the MLP model was trained on fasttext embeddings. Both lead to high R^2 and low MSE values. To verify that here, the metrics have an actual significance, Table 6 shows the prediction examples that could be seen earlier in Table 2 for both models.

phrase	SVR w2vec
very very happy	0.840
no support	0.754
was so pleased	0.747
quite understand	0.728
must be better	0.700
must accept	0.594
not too bad	0.259

phrase	MLP fasttext
very very happy	0.799
quite understand	0.699
must accept	0.653
was so pleased	0.643
must be better	0.575
not too bad	0.382
no support	0.318

Table 6: Gold Standard: Examples for Model Prediction on SCL-NMA

Especially the ranking of the MLP model is quite good. The SVR model seems to perform not as well as expected, but there seems to be a slight surprise when taking a look at the NP score prediction performance, which will be talked about in the next chapter.

4 Evaluation

4.1 NP Ranking

To rank the noun phrases and evaluate the final output, manual evaluation was needed. As annotating 8'000 NPs would be rather tiresome, I extracted 300 randomly for prediction and evaluation. The best models for each respective approach were used, namely the SVR word2vec silver standard model, the SVR word2vec gold standard model and the MLP fasttext gold standard model. As a bonus, the silver standard without any model was evaluated as well. Before I started the annotation task, I noticed that the outcome was rather surprising. In Table 7, the most positively ranked NPs from the sample with 300 noun phrases are shown. Even though the MLP model performed better than the SVR model on general phrases, this time, the SVR model seemed to produce a ranking of higher quality. This shows that a model with the ability to rank general phrases reasonably might not perform quite that well when it comes to noun phrases only. Thus, due to its low ranking quality, the MLP model was discarded from the manual evaluations.

Gold Standard: MLP fasttext	**Gold Standard: SVR word2vec**	**Silver Standard: SVR word2vec**
1985's Live Aid	a great admirer	a beautiful piece
Joy Division - Love	the Grammy Awards presentation	best musical score
two Olivier Awards	exclusive broadcasting rights	the Grammy Awards presentation
both Joy Division	All my best efforts	A few benefactors
Golden Globe awards	film chart crown	Spirit awards hail
- Best new musical - The Producers	Golden Globe awards	the marvellous, wonderful and talented Claire Kember
best musical score	her international career	the children's fiction award
Jude Law film Closer	the marvellous, wonderful and talented Claire Kember	best film nominees
An American version	best musical score	exclusive broadcasting rights
either Joy Division	the free party	flautist James Newton's Choir

Table 7: Rankings of NP Samples

4.2 Evaluation of the Approaches

4.2.1 Results

Approach	NPs with correct placement	Agreement
Silver Standard only	Annotator A: 0.913 Annotator B: 0.863	observed agreement: 0.896 Kappa: 0.482
Silver Standard as Training Data	Annotator A: 0.903 Annotator B: 0.857	observed agreement: 0.900 Kappa: 0.529
NRC-VAD as Training Data	Annotator A: 0.910 Annotator B: 0.867	observed agreement: 0.923 Kappa: 0.615

Table 8: Manual Evaluation of the Approaches

Table 8 shows us the results of the models judged by the manual annotation of two annotators. The annotators had to analyze 300 NPs in groups of 5 and pick out outliers that did not fit into the ranking within that window. They had to keep in mind that the NP on the top had to be the most positive, and the last one had to be the least positive within the respective group.

To make sure that the annotations are reliable, an agreement is measured with Cohen's Kappa. Only calculating the percentage of agreement would be misleading, as a percentage of agreements can happen by chance, and Kappa takes this into account (University of York Department of Health Sciences, 2005). Results within the range of 0.41 and 0.60 are considered moderate agreement according to Landis and Koch (1977), which is the case for all evaluations shown in the table. The possible reasons for these mediocre agreements will be discussed in the next subsection.

Regarding the percentage of correctly placed NPs, all three models were able to achieve a similarly high score. There is no notable difference and thus no approach that stands out. Interestingly, even the program for the silver standard generation without any model has no problems ranking NPs. Again, this shows that there seems to be little correlation when it comes to ranking general phrases like in the SCL-NMA dataset and ranking noun phrases, and might indicate that the former is a more complicated task.

4.2.2 Problems

After the annotations were done and ready to be evaluated, I noticed some problems with the way that everything was evaluated. The first problem was that there were cases, where the first annotator marked the exact opposite of the other annotator, as it can be seen in Figure 4. Technically, both can be right, as moving down the NP 'unpaid working experience' would lead to the same result as moving up the noun phrases 'the best documentary' and 'the biggest and most prestigious film event'.

1		Annotator A	Annotator B
51	her best actress nomination		
52	unpaid work experience	x	
53	the best documentary		x
54	the biggest and most prestigious film event		x
55	London's Tate Britain art gallery		

Figure 4: Annotation of NPs: Problem 1

Another issue is that some NPs can wrongfully be marked as correctly placed, only because of the coincidental split of the group at the right spot. An example is shown in Figure 5. There are two separate groups, and with each group, the positivity should decrease as everything grouped together is a ranking with decreasing polarity score. However, it makes a difference whether the second group is looked at in an isolated manner. If only group 2 is present, 'best rock vocal performance' is correctly placed. If we were to take group 1 into consideration as well, the situation would change, as phrases in group 1 mainly get across a neutral sentiment, while the 'best rock vocal performance' is quite a positive phrase and should be located far above.

63	One World Project
64	London's Tate Britain art gallery
65	a real person
66	16 million viewers
67	The plain green Norway spruce
68	
69	best rock vocal performance
70	the MOBO Awards
71	the silver screen
72	The "performance capture" technique
73	The Oscars nominations list

Figure 5: Annotation of NPs: Problem 2

This problem could be solved by creating an overlapped version of the groups, where the last one or two NPs of the group reoccur in the next group. While calculating the quality of the ranking, this could be taken into account by only counting the overlapping ones once, and always counting the misplaced version if there is a discrepancy between two versions. Whether this would actually help or create even more confusion when there are too many misplaced ones among the reoccurring, is another question. Overall, it seems that the safest way is to create a gold standard ranking with crowdsourcing and a method like BWS which was used to create the SCL-NMA in Kiritchenko et al. (2016).

4.3 Comparison with Results of SemEval-2016 Task 7

As the models were evaluated on general phrase rankings previously, they lend themselves to being evaluated in the same way as the models of the contestants in SemEval-2016 Task 7. The focus will be on one of the datasets only, namely the SCL-NMA.

The SemEval-2016 version of the lexicon has 2999 terms as stated in Kiritchenko et al. (2016), however, I was only able to find the lexicon with either 2799 terms on the SemEval-2016 resources website [1], or with 3207 on one of the author's website [2], which seems to be an updated version of the SCL-NMA lexicon. Thus, I included the results for both versions I was able to find in Table 9 and 10. The new lexicon leads to slightly better results than the one linked on the SemEval website in the multi-word phrase category, but other than that, there is not much difference.

The results of the teams are shown in Table 11. The approaches in this thesis and the second team appear comparable. While the model of the team UWB has a better score in the category of multi-word phrases, my models' single word scores are almost as high as the winning team's. In order to get an overall score like ECNU, the multi-word phrases need to be much higher. Investigating how to do this could be a task for future work.

[1] https://alt.qcri.org/semeval2016/task7/index.php?id=data-and-tools

[2] http://saifmohammad.com/WebPages/lexicons.html

Approach	**Overall**		**Single words**		**Multi-word phrases**	
	Kendall's τ:	Spearman's ρ:	Kendall's τ:	Spearman's ρ:	Kendall's τ:	Spearman's ρ:
Silver Standard only	0.541	0.696	0.702	0.871	0.391	0.520
SVR word2vec model trained on Silver Standard	0.549	0.747	0.644	0.850	0.437	0.620
SVR word2vec model trained on NRC-VAD	0.599	0.797	0.721	0.906	0.444	0.634
MLP fasttext model trained on NRC-VAD	0.620	0.821	0.721	0.905	0.459	0.658

Table 9: Results of Approaches on SCL-NMA (General English Sentiment Modifiers)

Approach	**Overall**		**Single words**		**Multi-word phrases**	
	Kendall's τ:	Spearman's ρ:	Kendall's τ:	Spearman's ρ:	Kendall's τ:	Spearman's ρ:
Silver Standard only	0.541	0.698	0.688	0.855	0.375	0.497
SVR word2vec model trained on Silver Standard	0.549	0.747	0.645	0.850	0.437	0.620
SVR word2vec model trained on NRC-VAD	0.590	0.789	0.721	0.906	0.444	0.634
MLP fasttext model trained on NRC-VAD	0.607	0.809	0.721	0.905	0.460	0.658

Table 10: Results of Approaches on SCL-NMA (SemEval-2016 Version)

Team	Overall		Single words		Multi-word phrases	
	Kendall's τ	Spearman's ρ	Kendall's τ	Spearman's ρ	Kendall's τ	Spearman's ρ
ECNU	**0.704**	0.863	0.734	0.884	0.686	0.845
UWB	0.659	0.854	0.644	0.846	0.657	0.849
LSIS	0.350	0.508	0.421	0.599	0.324	0.462

Table 11: Team Results on the SCL-NMA Lexicon (Kiritchenko et al., 2016)

5 Conclusion

In this project, I have shown several possible approaches to rank phrases and words according to their polarity values. Even though the systems are not perfect yet, they could provide a basis for future, better approaches. Especially the multi-word phrase predictions need work. Also, a better manual evaluation system is needed to carry out annotations reliably. For now, the best approach still seems to be creating a gold standard by using the BWS method. Furthermore, I now know that model evaluation metrics when calculated on models trained on a generated silver standard might not predict model quality reliably, as they are highly dependent on the silver standard quality.

All in all, the best ones among the models still seem to be satisfactory when compared to the results in Kiritchenko et al. (2016). Especially the SVR models appear to be well suited for polarity quantification tasks. Even with manual annotation, the ranking qualities are high. For future work, it would be interesting to see whether the same approaches work as well for other languages. As the NRC-VAD lexicon is also available for other languages, the gold standard approach would be a possibility to explore regarding that matter.

References

M. Awad and R. Khanna. *Support Vector Regression*, pages 67–80. Apress, Berkeley, CA, 2015. ISBN 978-1-4302-5990-9. doi: 10.1007/978-1-4302-5990-9_4. URL https://doi.org/10.1007/978-1-4302-5990-9_4.

S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly, Beijing, 2009. ISBN 978-0-596-51649-9. doi: http://my.safaribooksonline.com/9780596516499. URL http://www.nltk.org/book.

D. Greene and P. Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proc. 23rd International Conference on Machine learning (ICML'06)*, pages 377–384. ACM Press, 2006.

A. Göhring and M. Klenner. Polar Quantification of Actor Noun Phrases for German. Marseille, France, 2022. LREC.

M. Honnibal and I. Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

S. S. Keerthi and C.-J. Lin. Asymptotic behaviors of support vector machines with gaussian kernel. *Neural Computation*, 15:1667–1689, 2003.

M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938. ISSN 00063444. URL http://www.jstor.org/stable/2332226.

A. Kharwal. R2 Score in Machine Learning, June 2021. URL https://thecleverprogrammer.com/2021/06/22/r2-score-in-machine-learning/.

S. Kiritchenko, S. M. Mohammad, and M. Salameh. Semeval-2016 task 7: Determining sentiment intensity of english and arabic phrases. In *Proceedings of the International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, June 2016.

M. Klenner, A. Fahrni, and S. Petrakis. Polart: A robust tool for sentiment analysis. 05 2009. doi: 10.5167/uzh-19792.

K. Kumari and S. Yadav. Linear regression analysis study. *Journal of the Practice of Cardiovascular Sciences*, 4:33, 01 2018. doi: 10.4103/jpcs.jpcs_8_18.

J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1), 1977.

B. Liu. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167, May 2012. doi: 10.2200/s00416ed1v01y201204hlt016. URL `https://doi.org/10.2200/s00416ed1v01y201204hlt016`.

J. Martin and P. White. *The Language of Evaluation: Appraisal in English*. 01 2005. doi: 10.1057/9780230511910.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality, 2013. URL `https://arxiv.org/abs/1310.4546`.

T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

S. M. Mohammad. Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 english words. In *Proceedings of The Annual Conference of the Association for Computational Linguistics (ACL)*, Melbourne, Australia, 2018.

S. M. Mohammad and P. D. Turney. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29(3):436–465, 2013.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

R. Plutchik. Chapter 1 - a general psychoevolutionary theory of emotion. In R. Plutchik and H. Kellerman, editors, *Theories of Emotion*, pages 3–33. Academic Press, 1980. ISBN 978-0-12-558701-3. doi: https://doi.org/10.1016/B978-0-12-558701-3.50007-7. URL `https://www.sciencedirect.com/science/article/pii/B9780125587013500077`.

L. Polanyi and A. Zaenen. *Contextual Valence Shifters*, volume 20, pages 1–10. 01 2006. ISBN 1-4020-4026-1. doi: 10.1007/1-4020-4102-0_1.

L. Rosenbaum, A. Dörr, M. Bauer, F. Boeckler, and A. Zell. Inferring multi-target qsar models with taxonomy-based multi-task learning. *Journal of cheminformatics*, 5:33, 07 2013. doi: 10.1186/1758-2946-5-33.

S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press, 2014. ISBN 9781107057135. URL https://books.google.ch/books?id=ttJkAwAAQBAJ.

C. Spearman. The proof and measurement of association between two things. *International Journal of Epidemiology*, 39(5):1137–1150, 10 2010. ISSN 0300-5771. doi: 10.1093/ije/dyq191. URL https://doi.org/10.1093/ije/dyq191.

S. Srinivasan, editor. *Guide to Big Data Applications*. Springer International Publishing, 2018. doi: 10.1007/978-3-319-53817-4. URL https://doi.org/10.1007/978-3-319-53817-4.

M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede. Lexicon-Based Methods for Sentiment Analysis. *Computational Linguistics*, 37(2):267–307, 06 2011. ISSN 0891-2017. doi: 10.1162/COLI_a_00049. URL https://doi.org/10.1162/COLI_a_00049.

L. Tian, C. Lai, and J. D. Moore. Polarity and intensity: the two aspects of sentiment analysis, 2018. URL https://arxiv.org/abs/1807.01466.

University of York Department of Health Sciences. Measurement in Health and Disease: Assessing Agreement Using Cohen's Kappa, 2005. URL https://www-users.york.ac.uk/~mb55/msc/clinimet/week4/kappa.htm.

T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, page 347–354, USA, 2005. Association for Computational Linguistics. doi: 10.3115/1220575.1220619. URL https://doi.org/10.3115/1220575.1220619.