university of groningen

faculty of arts

Predicting aspects in Dutch book reviews using a dynamic lexicon

Jannick Akkermans

Bachelor thesis
Informatiekunde
Jannick Akkermans
s3429075

ABSTRACT

Reviews are one of the many ways in which people express their opinions about products and services. Generally, an overall rating is assigned to these reviews. These ratings hide a lot of underlying information such as descriptions of aspects about the product or service. Automatically identifying these aspects is called Aspect Identification. Most researches on this topic focused on explicitly mentioned aspects. Additionally, many similar types of research on Aspect Identification employ a linear classification model with Bag-of-Words features.

This thesis aims to improve the accuracy of such linear models by using information from POS tags, word embeddings and the genre of a book. This thesis focuses on sentences that are annotated with one label. Subsequently, the aspects of sentences are predicted based on features that are selected by a lexicon. The lexicon is created by using the information from POS tags and word embeddings. The process of creating the lexicon and selecting the features is conducted with the programming language Python. Each sentence is split up into tokens. Each token with a target POS is selected and the similarity between that word and the corresponding aspect is calculated. Once it exceeds the threshold of 0.2 (20%), it is recorded in a lexicon. Subsequently, each sentence is transformed into a sequence of zeros and ones based on whether a word occurred in the lexicon or not.

The results of this thesis show that the created classifier achieved an accuracy of 79%, therefore outperforming the baseline of 76.2%. Additionally, the classifier achieved an F1-score of 74%. Based on these results, it can be concluded that information from POS tags, word embeddings and the genre of a book increases the accuracy of a linear classification model. Although these results look promising, the data set used in this thesis was heavily unbalanced and many sentences were left unannotated. These are some issues that can be solved in future research.

CONTENTS

PREFACE

Right now you are reading the thesis 'Predicting aspects in Dutch book review using a dynamic lexicon'. This project started on the 3rd of February and lasted till the 5th of June. In the first week we already had to select our first choice for a topic. Almost all of the available topics looked interesting to me, but predicting aspects in book reviews seemed the most interesting to me. In the end, I'm glad I chose this topic.

Working on my thesis went better than I thought. I always had this idea that the thesis was a scary project and the programming part would go very badly. Actually, I was very wrong. The topic was really fun, the literature was easier to find than I thought, my seminar group was great and the programming part went really well. I could always find the right balance between working on my thesis and working on the other courses I followed.

I really want to thank my supervisor, Andreas van Cranenburgh, for guiding me through this process. He was always kind enough to help me with problems I encountered and his feedback was always super clear and really helped me to improve my thesis. I also want to thank my girlfriend and a good friend of mine for helping me with possible spelling errors, grammatical errors and making sure that the overall text is nice to read.

In the end, I hope that you enjoy reading this thesis on predicting aspects in Dutch book reviews using a dynamic lexicon.

Jannick Akkermans, Wolvega, 8th of May, 2020

1 INTRODUCTION

Have you ever written a review online about a laptop that you bought? Well, that is no surprise. Reviews are one of the many ways in which people can express their opinions about products and services. Generally, these reviews are assigned an overall rating based on stars. The issue with these ratings is that they hide a lot of underlying information (Xue et al., 2017). Such information could describe certain aspects about the product or service, for example the quality of the food when talking about a restaurant. Another problem with these overall ratings is that a reviewer's rating might reflect aspects in which a search user is not interested (Ganu et al., 2009).

Automatically identifying aspects in reviews is one of the tasks in Natural Language Processing (NLP). This technique is called Aspect Identification and it allows people to search more effectively for reviews mentioning the aspect of their interest. This will enhance the user experience in assessing reviews (Ganu et al., 2009). Aspect Identification is part of Aspect Based Sentiment Analysis (ABSA), which aims at identifying the aspects of entities and the sentiments expressed towards each aspect (Villaneau et al., 2018). An advantage of ABSA is that it can analyze large amounts of unstructured texts and extract information from it that is not included in the user ratings (Pontiki et al., 2016).

Current research in this field has applied ABSA in the English language combined with domains such as restaurants and laptops, in which the aspects are easy to define (Pontiki et al., 2016; Villaneau et al., 2018). Many approaches for finding these aspects have already been defined. However, most of these approaches focus solely on explicitly mentioned features, for example the quality of the food. This is due to the majority of features in consumer reviews being explicit (Schouten and Frascinar, 2014). On the other hand, aspects can also be mentioned implicitly. Automatically identifying these implicit aspects has proven to be more difficult, which is why it still remains a challenge in book reviews.

To this day, little research has yet been done on predicting implicit aspects from Dutch book reviews. Additionally, many similar researches on aspect detection employ a linear classification model with Bag-of-Words (BoW) features. Such models focus solely on the word frequencies from the training data. Therefore, this thesis will aim to assess whether information from word embeddings and Part-of-Speech (POS) tags can improve the accuracy of linear classification models. For this, the following question will be answered: 'How can the information from word embeddings and POS tags be of added value as opposed to a linear classification model?'. In addition, the genre of the book to which the review is directed is taken into account. Therefore, this thesis also aims to assess whether the genre improves the accuracy of linear classification models.

The remainder of this thesis is structered as follows: Chapter 2 describes previous techniques employed in the prediction of aspects in reviews. Chapter 3 elaborates on the data and materials used in this thesis. Chapter 4 discusses the method in which a detailed overview of the approach is provided. In chapter 5, the results of the described method are discussed and they will be critically evaluated. Finally, chapter 6 gives a final conclusion, as well as a discussion of the limitation and future work.

2 BACKGROUND

Aspect Identification as part of ABSA has been developed for a long time, starting in 2004 with the research of Hu and Liu (2004). Their research focused on summarizing product reviews based on features mentioned in the reviews. The features used in the reviews summarization are frequent features, i.e. features that are talked about by many customers. Hu and Liu (2004) only used features that appear as nouns or noun phrases in the reviews. However, all the features extracted from the reviews are explicitly mentioned, e.g. *The pictures are very clear*. All the implicitly mentioned features were left out of this research. The explicitly mentioned frequent features were subsequently pruned since not every feature was a genuine candidate. Therefore, Hu and Liu (2004) used two types of pruning: Compactness pruning and redundancy pruning. **Compactness pruning** checks whether features contain at least two words and removes those that are likely to be meaningless. **Redundancy pruning** removes redundant features that contain one word. To detect redundancy, each feature has an associated *p-support* value. If this value is less than the minimum p-support value and the feature is a subset of another sentence, it is pruned. An example of how redundancy pruning works is that *life* on its own is not a useful feature, but *battery life* is. The results showed that nouns significantly contribute to the detection of features. The results also showed that pruning significantly improves the results (Hu and Liu, 2004).

This research is useful for my thesis, as Hu and Liu (2004) showed that using nouns to retrieve features leads to relatively good results. In this thesis nouns will also be used, but now as features to predict aspects in book reviews. My thesis differs from this research because implicitly mentioned features are also taken into account. How they are retrieved from the reviews will be further explained in Chapter 4.

Another approach to predict aspects in reviews has been conducted by Hadano et al. (2011). They proposed an Aspect Identification technique for sentiment sentences in review documents. In order to gather training data for their classification model, a clustering method is used to group similar sentences together into clusters. The similar sentences are grouped together with the help of Bayon[1], which is a simple and fast hard-clustering tool. After the clusters are created, an annotator tags each cluster centroid with an aspect. Every sentence close to that centroid receives the same aspect. According to Hadano et al. (2011), this method is used since they assume that sentences in the same cluster are likely to elaborate on the same aspect. Subsequently, the sentences that are closest to their respective centroid, i.e. sentences with a similarity score in a defined range, are acquired as training data. This similarity score is based on the cosine similarity between the vector of the cluster centroid and the vector of the target sentence. An example of the form of sentence vectors in this research can be found in Appendix A. Ultimately, this new training data is used together with an SVM classifier to predict the aspects of sentences in the test data.

Using this clustering approach, Hadano et al. (2011) managed to achieve an accuracy of 73.97%. The training data that achieved this accuracy consisted of 250 sentences that had a similarity of 0.7-0.9 with their respective cluster centroid. This acquisition method, however, did not provide a large contribution since the model without the acquisition range achieved an accuracy of 73.80%. Hadano et al. (2011) stated that this is most likely due to the clustering process being based on Bag-of-

[1] https://code.google.com/archive/p/bayon/

Words (BoW) features. To achieve a higher accuracy, they propose that a method should be implemented that takes the semantic meaning of words into account.

In this thesis, the semantic meaning of words is used to derive better features for the prediction of aspects in book reviews. The details of this approach are further explained in Chapter 4. Similar to the research of Hadano et al. (2011), this thesis also employs a BoW-approach but with additional filters over the words.

The aforementioned researches have all focused on English reviews. De Clercq and Hoste (2016) are the first to have conducted research in ABSA using Dutch reviews. They created two distinct data sets of Dutch reviews from well-known domains: restaurants and smartphones. These data sets were subsequently used in the extraction of aspect expression terms. De Clercq and Hoste (2016) described expression terms as a word or words referring to a specific entity or aspect. Every candidate term was extracted from the reviews with the help of a reduced version of the hybrid terminology extraction system TExSIS (Macken et al., 2013) and annotated with the aspect it described. However, implicit expression terms were annotated as *NULL*. In their subsequent steps, only expression terms that contained subjectivity were used. This subjectivity was determined by a lexicon lookup. The lexicon was created using the LeTs Preprocess toolkit which linguistically preprocessed all the reviews (Van de Kauter et al., 2013). Additionally, semantic filtering was applied to filter out the expression terms that matched their respective domain. Semantic filtering was applied using Cornetto (Vossen et al., 2013) and DBPedia (Lehmann et al., 2015). The system of De Clercq and Hoste (2016), together with these filters, was ultimately tested against a held-out test set. On this test set, the system achieved an F-1 score of 46.24%.

In this thesis, the same technique will be used, namely applying a semantic filter over words. Therefore, the results of this research are highly useful. The semantic filter used in this thesis will not rely on Cornetto and DBPedia. Instead, it will be created using Fasttext, which creates word embeddings based on data that it has seen. Further details are explained in Chapter 4. Additionally, implicit expression terms will not be annotated as *NULL* in this thesis. Instead, every good candidate term, whether explicit or implicit, will be recorded in a lexicon.

Although the previous research yielded promising results and used Dutch reviews, it still used reviews from domains in which aspects are easy to define. Villaneau et al. (2018) are the first to conduct research in ABSA with book reviews. They collected a data set of 900 French book reviews. Each review was broken down in sentences and each sentence was transformed into a vector with binary entries. These entries code the co-occurrence of the sentence with a list of lemmas. The lemmas were nouns, verbs, adjectives and adverbs excluding stopwords. They were selected based on their frequency in the data set. The resulting co-occurrence matrix was then augmented with a column that specified the aspect and attribute that were assigned to the sentence. An example of the form of the co-occurrence in this research can be found in Appendix A. Using this co-occurrence matrix, Villaneau et al. (2018) build several systems of which Random Forest, Support Vector Machine and Support Vector Machine with Word2Vec achieved the most promising results. Combining the decisions of these three systems, however, achieved even better results. The final system achieved a macro F-1 score of 70.50%.

Techniques in this research will also be implemented in my thesis. For example, I also plan to use the POS tags of words to select nouns, verbs, adjectives and adverbs. Stopwords will be included in my research since they could still provide useful information for classification and house sentiment information. Additionally, this thesis will also focus on Aspect Identification in book reviews, but in the Dutch language. How this thesis also differs from Villaneau et al. (2018), is that they used the semantic information of words with the help of Word2Vec. My research will use word embeddings from Fasttext instead. In Chapter 4, I will explain why Fasttext is used as opposed to Word2Vec.

3 DATA AND MATERIAL

3.1 COLLECTION

The data that will be used for this thesis contains 3000 sentences from Dutch book reviews. These sentences are extracted from the review site Hebban.nl and are stored inside a CSV file called **review_data_v6.csv**. Each line in this CSV file consists of several columns. The first four columns contain labels that have been assigned to that sentence by two annotators. The reason for two annotators to annotate a sentence will be explained in section 5.1. The fifth column contains the review sentence. The sixth column contains the name of the file in which this sentence was stored. Lastly, the seventh column contains the ID of the sentence which specifies the exact position of that sentence in the review.

In order to assign the POS tags to all the sentences, a data set with parsed sentences is used that corresponds to the sentences in **review_data_v6.csv**. These sentences come from the full book reviews data set, which contains 100 thousand Dutch book reviews divided among 383 files. All these reviews are split up into sentences and all the sentences are parsed according to the guidelines of the Universal Dependencies (UD). UD is a framework for consistent annotation of grammar across different human languages[1]. Every parsed sentence is preceded by several tags, such as *sent_id*, which contains the sentence ID corresponding with the ones in the aforementioned CSV file. After these tags, several word lines follow depending on the number of tokens in the sentence. Each of these word lines contains the following fields:

1. Token index
2. The token itself
3. The lemma of the token
4. The POS-tag of the token
5. A Dutch translation of the POS-tag
6. A list of morphological features that the token possesses
7. The head of the current token, which is either a value of the token index or zero.
8. Universal dependency relation to the head
9. An enhanced dependency graph in the form of a list of head-deprel pairs
10. Any other annotation

The *sent_id* tag will be used to link the parsed sentence to the corresponding sentence in **review_data_v6.csv**. After the link is made, the POS-tag will be assigned to each word.

For each sentence in the previously mentioned CSV file, the genre of the book that the sentence describes will be added. This metadata is stored inside the text-file **odbrdata.txt**. Each line in this file contains several metadata items separated by tabs. These metadata items are:

[1] https://universaldependencies.org/

1. The URL of the review the sentences is derived from
2. The ID of the account that posted the review
3. The date on which the review was posted
4. The rating that the reviewer gave the book
5. The ID of the book which the review describes
6. The NUR-code of the book
7. The ISBN code of the book
8. The author of the book
9. The title of the book

Amongst this metadata, the NUR-code is the most important one. This code corresponds to the genre of a book, e.g. 305 stands for a literary thriller. Through the sentence ID field in **review_data_v6.csv**, the review sentence will be linked to the corresponding metadata in **odbrdata.txt**. If a review sentence for example has a sentence ID of 4806_1.p.1.s.9, then it will first be linked to the 4806th line in the url file **urls.txt**. Then the corresponding url in **odbrdata.txt** is retrieved. From that line, the NUR-code will be extracted and that code is then assigned to the review sentence.

3.2 ANNOTATION

A required component for the training data are the training labels. These labels are provided by means of annotating the review sentences in **review_data_v6.csv**. Each sentence will be manually annotated with 0, 1 or 2 overall labels which consist of two parts. The first part is a number (1-5) that refers to a specific aspect as defined in Table 1. The second part is a sentiment label that can either be + for positive, - for negative or empty for neutral. When a sentence contains no aspect or sentiment, whether explicit or implicit, it will receive no label. In this thesis, only sentences that contain one overall label are used. This reduces the size of the data set to 2000-2500 sentences. In addition, selecting sentences with only one label results in an easier classification task. This could produce higher results than a classification task in which sentences with two labels or no labels are also used.

The aforementioned annotation is based on existing guidelines for annotating book reviews[2]. However, these guidelines comprise more than 12 different aspects. Therefore, they are revised in order to comprise the five aspects that are described in Table 1.

[2] https://pboot.github.io/bookcrit/

Label	Aspect	Description	Sample words
1	Style	Discussion of tone and language use in the work, w.r.t. word choice, syntax and its connotations	stijl, taal, toon, ondertoon, penvoering
	Structure	Discussion of the macrostructure of the text, relations between the chapters or other textual components of the text	structuur, orde, ordeloosheid, sprongen in ruimte of tijd, vermenging
2	Plot	Discussion of events and ending of the story	plot, verhaal, verhaaldraad, slot, ontknoping
	Theme	Discussion about the topic that are written about	thema, idee, probleem, problematiek, gedachte
3	Characters	Discussion of characters in the book	personages, figuur, hoofdpersoon, karakter, mens
	Dialogue	Discusses how conversations between characters are represented	dialoog, gesprek, woordenwisseling, uitspraak, vertellen, spreken
4	Appearance	Discussion of paratextual and non-textual data, such as title, cover, publisher, genre indication, etc	uitgever, titel, illustratie, afbeelding, foto, flaptekst
5	Entire work	Discussion of the book as a whole or of parts of the book, where no reference is made to specific aspects	boek, werk, roman, verhaal, vertelling, deel, hoofdstuk, slot

Table 1: Overview of the annotation labels along with descriptions and examples of sample words

3.3 PROCESSING THE SENTENCES

3.3.1 Processing the data from review_data_v6.csv

To train the classifier, features have to be extracted from the data. How the features will be derived from the sentences is described in Chapter 4. Before the features can be extracted, the data first needs to be cleaned with the help of pre-processing. First of all, the sentences are tokenized. Tokenization is the process of splitting a sentence into tokens, or basic units that need no further decomposition (Webster and Kit, 1992). In this thesis, the tokenization step will solely involve splitting the sentence on the space character, because the sentences were already tokenized when they were extracted from Hebban.nl. The next step is lowercasing each token to remove duplicate tokens and therefore reduce the number of unique tokens. This results in a smaller number of features which is usually more beneficial in classification tasks. The last step of the pre-processing is the removal of punctuation. I remove punctuation from the sentences, because they do not provide useful information for the classification of aspects.

3.3.2 Processing the 110k book reviews

As mentioned earlier, the POS tags will be derived from the full book reviews data set, which contains 100 thousand Dutch book reviews. These reviews are divided among 383 files and they are stored in the CoNNL-U format, which is the standard for storing UD-parsed data. In section 3.1, I described how each sentence is represented in any of these files. To process this data set, the conllu parser is used in Python. Each file in the data set will be processed with the function *parse_incr()* which can parse many sentences without loading them all into memory at once. After a sentence is parsed, it contains two parts. One part contains the metadata, which are the tags described in section 3.1. The other part contains a token list in which for every token in the sentence, a dictionary is created containing the 10 fields described in section 3.1. For each sentence parsed by the conllu parser, the sentence ID will be derived from the metadata. This ID is accompanied by a list of tuples, i.e. (*token*, *POS*) pairs. These are derived by means of extracting the token and POS-tag from the token dictionary for each token in the token list. This combination of sentence ID and list of tuples is then written to a text file called **processed_output_v2.txt**.

4 METHOD

This thesis focuses on the classification of aspects in Dutch book reviews. To execute the classification and the necessary processing of the data, the programming language Python will be used. Along with Python, I will also use the library NLTK.

4.1 CONSTRUCTION OF THE CLASSIFIER

4.1.1 Reading in data

First of all, the data from **review_data_v6.csv** is read in. Each sentence is subsequently pre-processed using the steps explained in section 3.3. The processed sentence is then stored inside a dictionary, in which the sentence ID comprises the key and the corresponding value consists of a sub-dictionary. Each key in this sub-dictionary will be the column name of the remaining columns described in section 3.1 and the values will be the corresponding column values.

The next file that is read in, is **processed_output_v2.txt**. This file contains the processed data of all the one hundred thousand book reviews in one dictionary. It had to be converted into a string when it was stored inside the txt-file. To transform it back into the original dictionary, the function *json_loads()* from the JSON package is used.

The last file that will be read in, is **odbrdata.txt**. This file contains the metadata corresponding to each sentence in **review_data_v6.csv**. The metadata is read in using the function *DictReader()* from the CSV package. This function transforms each line into a dictionary in which the keys are the column names and the values are the corresponding values.

4.1.2 Generation of the lexicon

Another required component for classification is a set of features associated with a label. The features will be selected based on a generated lexicon. At the start, this lexicon contains seed words for each of the defined aspects that are likely to imply their respective aspects. Other words that imply certain aspects will be derived from the pre-processed sentences. Since not all words are equally useful for the classification, some sort of filter will be applied. This involves assigning POS tags to the words. These tags will be derived from **processed_output_v2.txt**. Villaneau et al. (2018) identified nouns, verbs, adjectives and adverbs as prime indicators for aspects. Therefore, I will focus on these POS tags for the construction of the lexicon.

The procedure for the construction of the lexicon is outlined in Figure 1. The similarity score mentioned in Figure 1 will be calculated by using word embeddings, constructed by the algorithm FastText. Word embeddings allow words with a similar meaning to be represented similarly. Fasttext will create word embeddings based on the full book reviews dataset. The advantage of such in-domain word embeddings over the ones trained on Wikipedia or news articles is that specific characteristics of the domain can be captured, such as domain-specific connotations. When the calculated similarity score exceeds a certain threshold, it can be assumed that that specific word is related enough to the respective aspect to be informative for the classification. The optimal threshold will be determined through experimentation. When the threshold is exceeded, the word will be recorded in the

```
translation dictionary = {'1': ['stijl', 'structuur'],
                          '2': ['plot', 'thema'],
                          '3': ['karakters', 'dialoog'],
                          '4': 'verschijning',
                          '5': 'gehele werk'}

for each sentence in the data set:
    for each word in the current sentence:
        if POS of word in ['noun', 'verb', 'adverb', 'adjective', 'proper noun']:
            if word already in feature lexicon:
                move to next word
            else:
                derive Dutch translations of aspects from translation dictionary
                based on the aspect number of the current sentence
                for each derived aspect:
                    calculate similarity score between that word and the derived aspect
                take the maximum similarity score from the calculated similarity scores
                if maximum similarity above threshold:
                    record the word in the feature lexicon
                else:
                    move to next word
        else:
            move to next word
```

Figure 1: Generation of the lexicon in pseudocode

lexicon. This process continues until all the sentences are processed and the lexicon is finished.

4.1.3 Training the classifier

When the pre-processing and construction of the lexicon are done, each sentence will be transformed into a set of feature values. When a word in a given sentence does not occur in the set of implication words for a given aspect in the lexicon, it will be added with a 0 to the set of features. When a word does occur in the lexicon, it will be added with a 1 to the set of features. This process is called sentence vectorization and transforms a sentence into a set of 0's and 1's. The genre of the book which is discussed in the sentence is also added with a 1 to the set of features. Figure 2 shows an example set of feature values. The sentence that is used in the example is *Het was een goed geschreven boek*.

```
({'het': 0, 'was': 0,
'een': 0, 'goed': 0,
'geschreven': 0, 'boek': 1,
'nur_305': 1}, 5)
```

Figure 2: Example of the resulting sentence representation

All the sentences with their labels will then be divided into a training set, a development set and a test set according to the 80-10-10 scheme. This means that at random, 80% of the data will be assigned to the training set, 10% of the data to the development set and 10% of the data to the test set. The training set is used to train a classifier. For this research, a Support Vector Machine (SVM) classifier with a linear kernel is used. This is the most suitable option when working with textual data since this kind of data is unstructured and contains a large number of potentially relevant features.

4.2 EVALUATION OF THE RESULTS

After the classifier has been trained on the training set, it will be evaluated on the development set. The development set serves as a substitute test set during the development of the classification system. When certain parameters or features in the system are changed, it will be trained on the training set again and subsequently tested on the development set. Only when the results on the development set are sufficient, will the system be tested on the final test set.

When the trained classifier is tested on the development or test set, it will assign a label to each set of features. This given label corresponds to the labels in Table 1. The labels are the predictions of the classifier according to patterns it has seen in the training data. These predictions will be used to evaluate the quality of the classification. The quality is assessed in terms of comparing the predictions of the classifier against the gold standard, which was manually annotated. This comparison results in four different metrics, namely the accuracy, precision, recall and F1-score. These metrics will be reported for each of the labels in Table 1. Additionally, an overall accuracy score based on the classification results of all the labels is reported. The calculations for the aforementioned metrics are executed using NLTK.

The overall F1-score is the most important one of the resulting measures, as this will ultimately be used to compare the classification used in my thesis against the baseline, which is derived from SemEval-2016 Task 5: Aspect Based Sentiment Analysis (Pontiki et al., 2016). This shared task of 2016 featured data sets from various domains, such as restaurants and laptops, in multiple languages including Dutch. The shared task consisted of several subtasks and slots and participants were free to choose in which subtask and slot they wanted to participate (Pontiki et al., 2016). Out of all the available subtasks and slots, subtask 1 in combination with slot 1 is the most relevant to use as a baseline since this dealt Aspect Identification on the sentence level. In this task, an SVM classifier with a linear kernel was used. The features used to train the classifier were unigram features extracted from the sentences in the training data. As previously mentioned, this is also my proposed method. However, I will combine this with information from word embeddings, which are trained on a different domain (books) and POS tags.

The best system in this task achieved an F1-score of 60.2%. The dataset that was used contained one sentence per review just like my dataset. One slight difference between the SemEval-2016 task and my research is that they used restaurant reviews whereas I use book reviews. As mentioned in the introduction, aspects are more difficult to detect in book reviews since they are often implicit. Nevertheless, the similarity in the design of the research makes it easier to compare the results and to answer the question whether information from word embeddings and POS tags can be of added value to the classification of aspects in Dutch book reviews.

All the code and necessary data can be found on my GitHub page. Additional files that are needed can be downloaded from my Google Drive.

5 RESULTS AND DISCUSSION

5.1 STRENGTH OF THE ANNOTATION

To be able to put the classification results into perspective, the quality of the annotation guidelines has to be assessed. This can be evaluated by calculating the agreement between multiple annotators, also known as the Inter-Annotator Agreement (IAA). Since one review can receive at most two labels, the IAA is calculated with the function *masi_distance()* from NLTK. This function takes partial agreements into account, as opposed to standard calculations of the IAA. In addition, *mase_distance()* calculates the distance between the annotations of the two annotators. Subtracting this score from 1 yields the agreement for a certain sentence.

When this function is applied to our dataset, it yields an IAA of 0.41. According to the guidelines of Landis and Koch (1977), this means we achieved a moderate agreement level. This score is supported by the IAA scores per annotator pairs, since these are 0.46, 0.46, 0.36, 0.47, 0.27 and 0.38 respectively. The average IAA score shows that there is still room for improvement in the annotation, even though we achieved a moderate agreement.

The agreements between two annotators for label 1 and label 2 separately can be found in Appendix C. The agreements between each pair of annotators can be found in Appendix D.

5.2 RESULTS OF THE CLASSIFIER

During the development of the classifier, it was trained on the training set and tested on the development set. To establish the influence of the POS tags and word embeddings, the classifier was build in several steps. Each of these implemented steps and its influence on the results is presented in Table 2:

Table 2: Results of the added features on the development set

Feature added	Accuracy	F1-score
Baseline (simple BoW-model)	0.50	0.41
NUR-code	0.50	0.43
POS tag filter	0.52	0.43

Lastly, the word embeddings filter was applied to arrive at the sets of features as described in Chapter 4. By training the classifier on the resulting features and testing it on the development set, it achieved the following results:

Table 3: Results of the SVM classifier on the development set.

Aspect	Accuracy	Precision	Recall	F1-score
Overall	0.76	0.75	0.71	0.72
1	0.71	0.77	0.71	0.74
2	0.77	0.81	0.77	0.79
3	0.73	1.00	0.73	0.85
4	0.45	0.50	0.45	0.48
5	0.88	0.68	0.88	0.76

These results show that in the context of aspect classification in book reviews, a model that uses information from POS tags and word embeddings scores better than a linear model that uses a BoW-approach. The results in Table 3 were achieved with an SVM classifier in which the following settings were used: C=1, kernel='linear'. The threshold value (v) for the word embeddings was 0.2. All the combinations tested can be found in Appendix B. In the appendix, it can be seen that a threshold value of 0.1 provides better results. However, with these higher results, there is also a higher chance that the classifier might overfit on the training data. This lowers the scores that it would achieve on the final test set. For this reason, I chose the threshold value of 0.2.

With the aforementioned settings, the classifier was tested on the final test set. At this stage, I combined the training and development data since more data usually means that a classifier performs better. This is further supported by Figure 3. As the amount of training data increases, the accuracy achieved on the development set increases. This hints that with more training data, my classifier could achieve higher results. For that reason, the training data and the development data are combined.

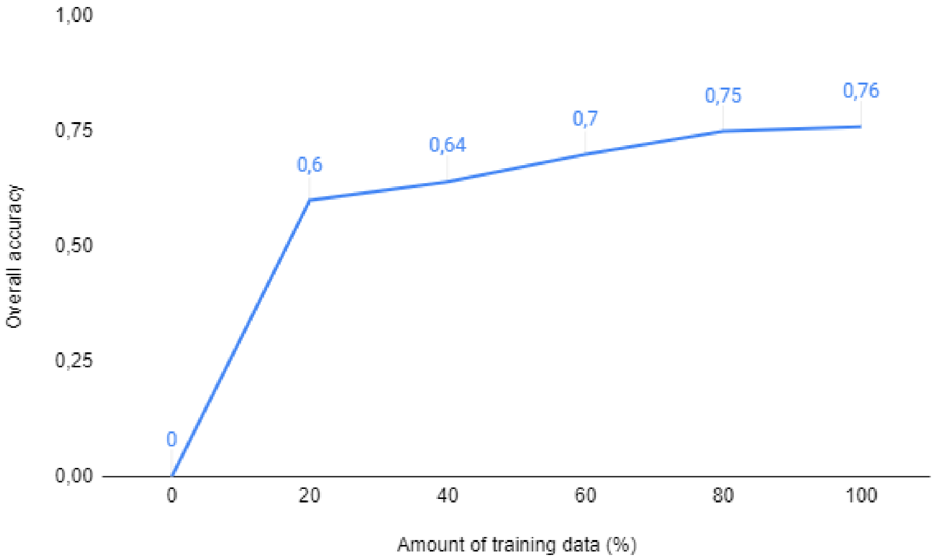


Figure 3: Results achieved on the development set with increasing amounts of training data

By combining the training data and the development data, the classifier was trained on 90% of the data, which is about 1500 sentences, and tested on the final 10%, which is 167 sentences. Ultimately, the classifier achieved the following results on the test set:

Table 4: Results of the SVM classifier on the final test set.

Aspect	Accuracy	Precision	Recall	F1-score
Overall	0.79	0.83	0.74	0.77
1	0.57	0.76	0.57	0.65
2	0.74	0.85	0.74	0.79
3	0.75	0.95	0.75	0.84
4	0.67	0.91	0.67	0.77
5	1.00	0.68	1.00	0.81

This means that in 79% of the cases, my classifier predicted the correct aspect. To further understand these results, a confusion matrix was created to show what kind of mistakes the classifier made:

		Gold aspect				
		1	2	3	4	5
Predicted aspect	1	**13**	6	0	0	4
	2	3	**45**	1	0	12
	3	1	2	**21**	1	3
	4	0	0	0	**10**	5
	5	0	0	0	0	**51**

The most notable mistake is the misclassification of 'plot/theme' when it should have been 'entire work'. This could be due to the fact that when someone describes an entire book, they could also make references to the plot. In such cases, the classifier could mistake the sentence for describing the plot of the book as these aspects show more or less similar words.

Overall, the final classifier scored better than the baseline in the same context. This means that my classifier is better at predicting aspects in book reviews than the baseline. Additionally, the scores on the test set are higher than on the development set, which indicates that my classifier has not overfitted on the development data.

5.3 COMPARING AGAINST RELATED WORK

The baseline that was selected for this thesis came from SemEval-2016 Task 5: Aspect Based Sentiment Analysis (Pontiki et al., 2016). This task involved a subtask in which participants trained and tested an SVM classifier on restaurant reviews to predict the aspects of these reviews. The features used in the classifiers were unigram words. As earlier described, the best system achieved an F1-score of 60.2%. To put the results of my classifier into the same context as the results of Pontiki et al. (2016), both studies need to be compared. Both studies use reviews containing only one sentence and we are both trying to predict one aspect. The same classifier is used in both studies and the same pre-processing steps are applied to the data. However, Pontiki et al. (2016) used restaurant reviews whereas I used book reviews. Unlike restaurant reviews, aspects are more implicit in book reviews and therefore they are more difficult to predict. Still, due to the similarities in research design, my research is comparable to the results of Pontiki et al. (2016). Given that my final system achieved an F1-score of 0.77, it can be said that my classifier scored better than a linear classification model and that the information from POS tags and word embeddings has a positive influence.

6 CONCLUSION

This thesis aimed at evaluating whether information from word embeddings, POS tags and the genre of a book can improve the accuracy of a linear classification model. The results showed that a model in which these techniques are used outperformed a linear classification model that uses a simple BoW-approach. Additionally, the created classifier showed promising results for future work. It achieved an accuracy of at least 0.6 and an F1-score of at least 0.7 on every aspect except for aspect 'style/structure'. This could be because some sentences annotated with a specific label any words in the lexicon for that label. This results in a sentence that is represented solely by zero's. Even though the ultimate threshold value for the similarity between words was very low (0.2), such cases still could have been present in sentences with label 'style/structure', 'plot/theme' and 'entire work', resulting in a misclassification rate of nearly 50% for label 'style/structure'. Future work could investigate other methods for establishing similarity between a word and an aspect which result in more words that implicate certain aspects.

For the word embeddings used in this thesis, I used FastText. This algorithm has several parameters that could improve the vector-space model that FastText creates, such as the vector dimension (*size*) and context window size (*window*). The vector dimension is the size of the learned word vector and the context window size is the range of words selected as the context for a target word (Chiu et al., 2016). Increasing these parameters captures more accurate word representations, but is more computationally costly. I used a *size* of 100 and a *window* of 10. Future work could increase these parameters with the hope of achieving higher results.

A major drawback in this research was the content of the data set. Although I used slightly more than 50% of all the review sentences, there were a significant number of sentences that received no labels as seen in Figure 4 in Appendix C. This could have happened because the annotation guidelines were inclusive enough or because aspects were not present enough to annotate the given sentence. For example, if a review described the story of a book it could be assigned the label 'plot/theme'. However, the annotation guidelines did not specify that such cases should be annotated with label 'plot/theme'. Therefore, the annotators did not annotate such sentences. This disagreement is also one of the most noticeable mistakes in the confusion matrices in Appendix C and Appendix D.

If the sentences with no labels were annotated, I could have had a bigger data set to train the classifier on. This could have yielded better results in the end. The disagreement of no label or an aspect label between the annotators is also the decision that has the largest number of mistakes. Appendix C and Appendix D show that this mistake happens more often than other types of mistakes. This too could be due to unclear annotation guidelines or due to the subjectivity of the annotators. However, compared to the number of agreements that a sentence should not receive a label, it is likely that these sentences in fact should not receive any labels.

Additionally, the distribution of labels was highly unbalanced. Label 'Appearance', for example, was represented very little whereas many sentences were annotated with label 'plot/theme'. This could also be due to unclear annotation guidelines or the subjectivity of the annotators as supported by the Inter-Annotator Agreement value of 0.41. Therefore, collecting more data or making the annotation guidelines more detailed is a note for future work. Additionally, annotators could meet more to resolve issues in the annotation. This would increase the Inter-Annotator Agreement, therefore providing a more solid annotation.

Future research could also make use of the sentences annotated with multiple labels. Since my thesis focused on sentences that were annotated with only one label, many sentences with two labels were left out which made my classification task easier compared to a classification task that would also try to predict the first or both of the labels. This is most likely the reason that my classifier produced the high results shown in Table 4. Including the sentences with two labels results in a more difficult classification task. This could potentially produce lower results.

All in all, the research question in this thesis can be answered. The information from POS tags, word embeddings and the genre of a book increases the accuracy of linear classification models. Although the results look promising, more research and more data are required to achieve higher results.

BIBLIOGRAPHY

Billy Chiu, Gamal Crichton, Anna Korhonen, and Sampo Pyysalo. 2016. How to train good word embeddings for biomedical nlp. In *Proceedings of the 15th workshop on biomedical natural language processing*, pages 166–174.

Orphée De Clercq and Véronique Hoste. 2016. Rude waiter but mouthwatering pastries! an exploratory study into dutch aspect-based sentiment analysis. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2910–2917.

Gayatree Ganu, Noemie Elhadad, and Amelie Marian. 2009. Beyond the stars: Improving rating predictions using review text content. In *Proceedings of WebDB*, volume 9, pages 1 – 6, Providence, Rhode Island, USA.

Masashi Hadano, Kazutaka Shimada, and Tsutomu Endo. 2011. Aspect identification of sentiment sentences using a clustering algorithm. *Procedia-Social and Behavioral Sciences*, 27:22–31.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Marjan Van de Kauter, Geert Coorman, Els Lefever, Bart Desmet, Lieve Macken, and Véronique Hoste. 2013. Lets preprocess: The multilingual lt3 linguistic preprocessing toolkit. *Computational Linguistics in the Netherlands journal*, 3:103–120.

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

Lieve Macken, Els Lefever, and Veronique Hoste. 2013. Texsis: Bilingual terminology extraction from parallel corpora using chunk-based alignment. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, 19(1):1–30.

Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeny Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of SemEval-2016*, pages 19 – 30.

Kim Schouten and Flavius Frascinar. 2014. Finding implicit features in consumer reviews for sentiment analysis.

Jeanne Villaneau, Stefania Pecore, and Farid Said. 2018. Aspect detection in book reviews:experimentations. In *Natural Language for Artificial Intelligence 2018*, volume 2244, pages 16 – 27.

Piek Vossen, Isa Maks, Roxane Segers, Hennie van der Vliet, Marie-Francine Moens, Katja Hofmann, Erik Tjong Kim Sang, and Maarten de Rijke. 2013. Cornetto: a lexical semantic database for dutch. in peter spyns et al., editors. *Essential Speech* *and Language Technology for Dutch, Theory and Applications of Natural Language Processing*, pages 165–184.

Jonathan J Webster and Chunyu Kit. 1992. Tokenization as the initial phase in nlp. In *COLING 1992 Volume 4: The 15th International Conference on Computational Linguistics*.

Wei Xue, Tao Li, and Naphtali Rishe. 2017. Aspect identification and ratings inference for hotel reviews. In *World Wide Web (2017)*, volume 20, pages 23 – 37.

A APPENDIX A

A.1 EXAMPLE FROM HADANO ET AL. (2011)

Table 5: Example of two sentence vectors with a cosine similarity of 0.822

Sentence	me	Jane	Julie	Linda	likes	loves	more	than
1	1	0	1	1	0	1	1	1
2	1	1	1	0	1	1	1	1

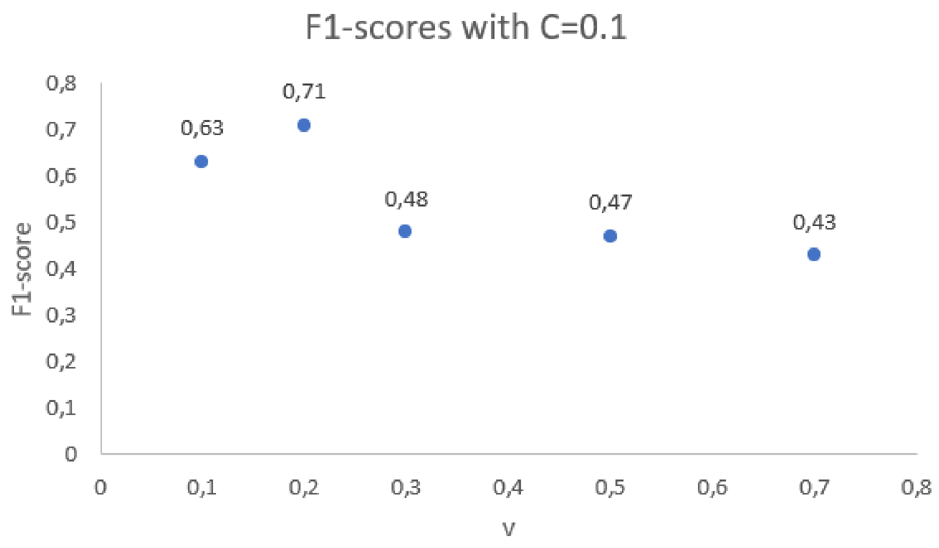
A.2 EXAMPLE FROM VILLANEAU ET AL. (2018)

Table 6: Example of a co-occurence matrix from the research of Villaneau et al. (2018)

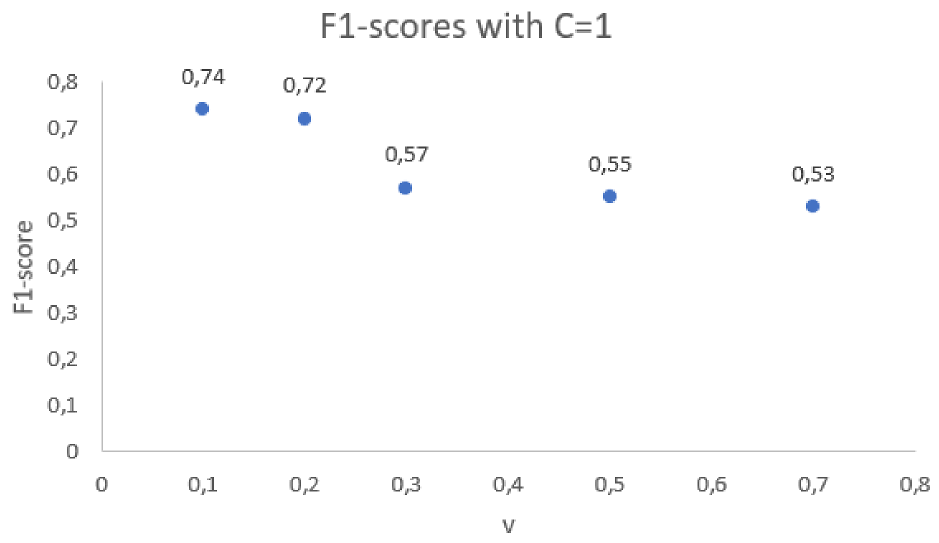
	Lemma 1	Lemma 2	Lemma 3	Lemma 4	Expression
Sentence 1	1	0	0	1	Text#Narrative
Sentence 2	0	0	1	1	Text#Style
Sentence 3	0	1	1	0	Author#Age

B INTERMEDIATE CLASSIFIER RESULTS

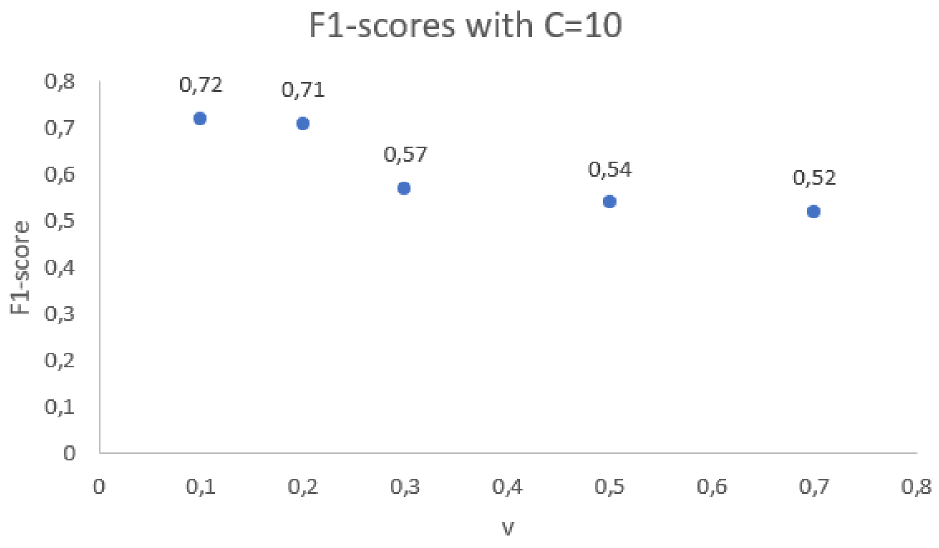
B.1 RESULTS WITH C=0.1



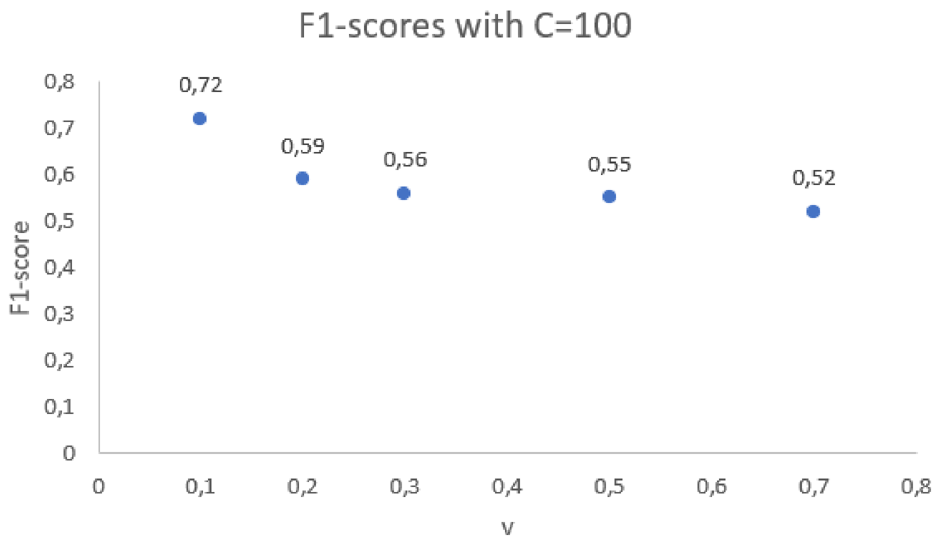
B.2 RESULTS WITH C=1



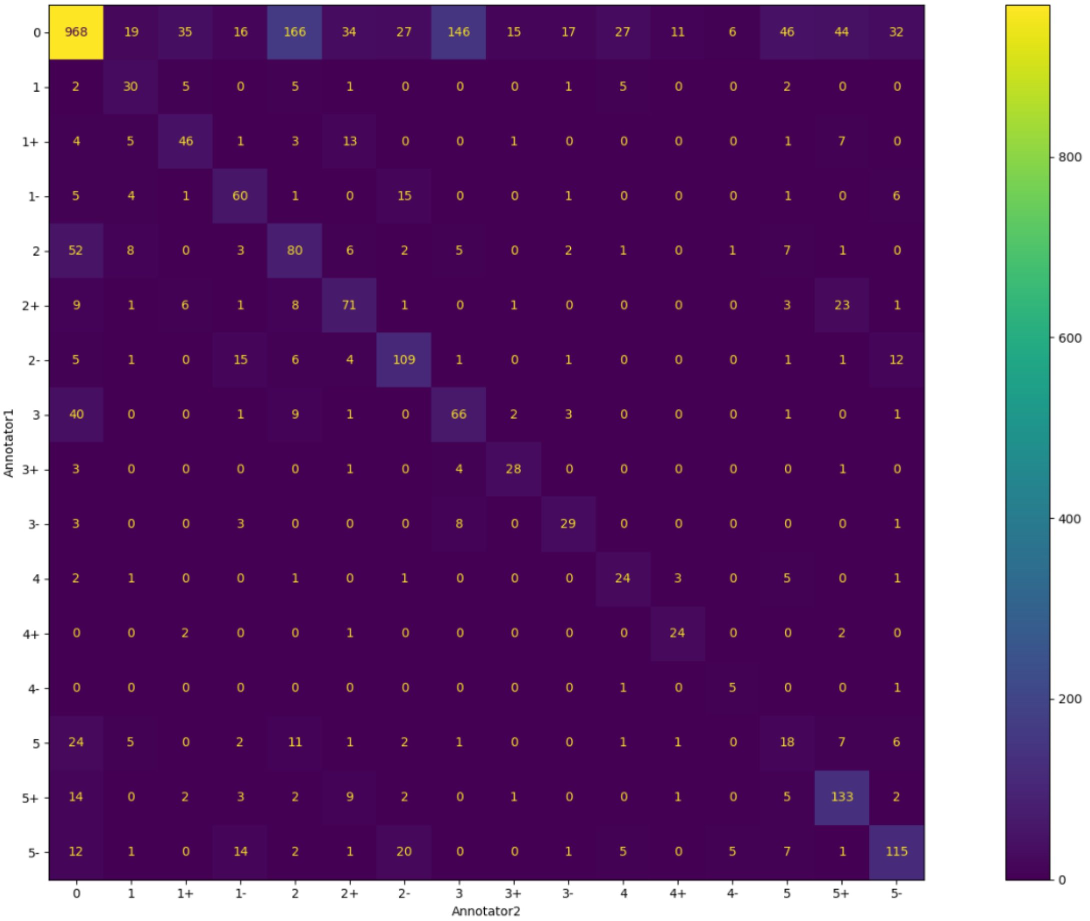
B.3 RESULTS WITH C=10



B.4 RESULTS WITH C=100

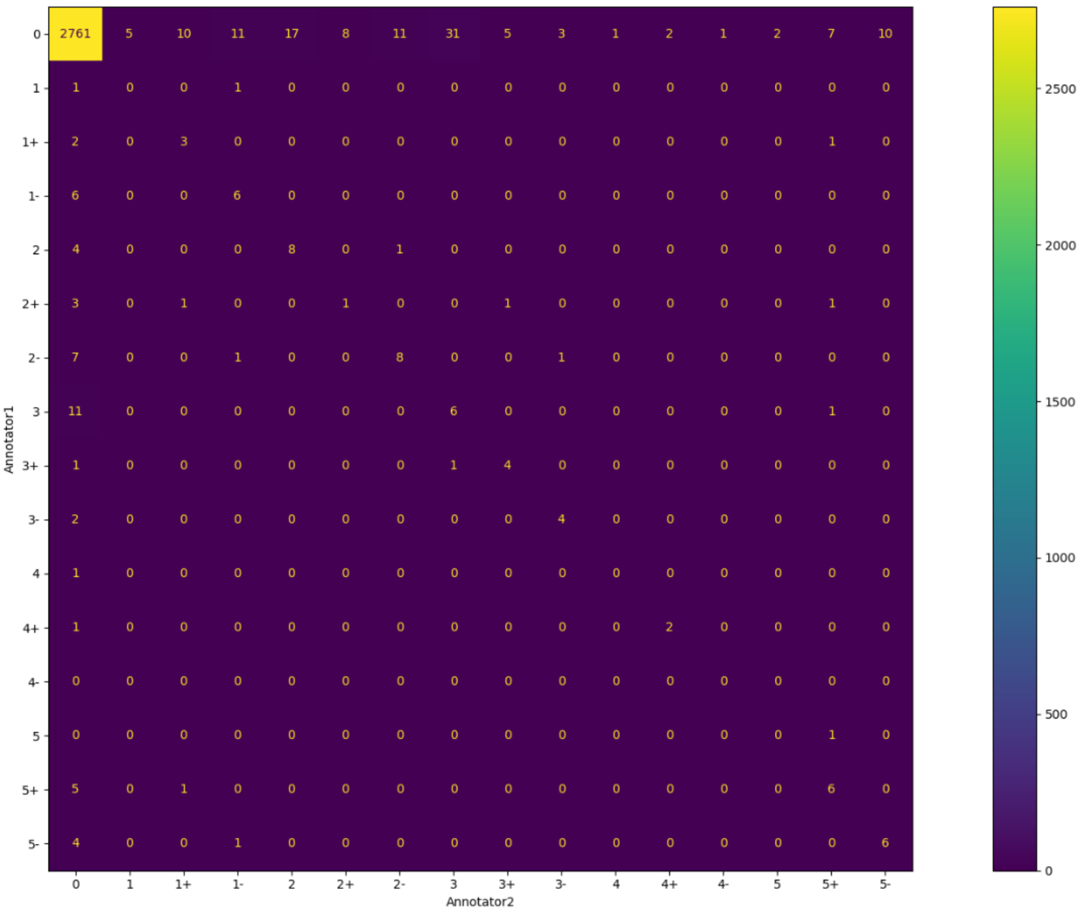


C CONFUSION MATRICES PER LABEL



	0	1	1+	1-	2	2+	2-	3	3+	3-	4	4+	4-	5	5+	5-
0	968	19	35	16	166	34	27	146	15	17	27	11	6	46	44	32
1	2	30	5	0	5	1	0	0	0	1	5	0	0	2	0	0
1+	4	5	46	1	3	13	0	0	1	0	0	0	0	1	7	0
1-	5	4	1	60	1	0	15	0	0	1	0	0	0	1	0	6
2	52	8	0	3	80	6	2	5	0	2	1	0	1	7	1	0
2+	9	1	6	1	8	71	1	0	1	0	0	0	0	3	23	1
2-	5	1	0	15	6	4	109	1	0	1	0	0	0	1	1	12
3	40	0	0	1	9	1	0	66	2	3	0	0	0	1	0	1
3+	3	0	0	0	0	1	0	4	28	0	0	0	0	0	1	0
3-	3	0	0	3	0	0	0	8	0	29	0	0	0	0	0	1
4	2	1	0	0	1	0	1	0	0	0	24	3	0	5	0	1
4+	0	0	2	0	0	1	0	0	0	0	0	24	0	0	2	0
4-	0	0	0	0	0	0	0	0	0	0	1	0	5	0	0	1
5	24	5	0	2	11	1	2	1	0	0	1	1	0	18	7	6
5+	14	0	2	3	2	9	2	0	1	0	0	1	0	5	133	2
5-	12	1	0	14	2	1	20	0	0	1	5	0	5	7	1	115

Figure 4: Confusion matrix of the agreement between annotator 1 and annotator 2 for label 1



	0	1	1+	1-	2	2+	2-	3	3+	3-	4	4+	4-	5	5+	5-
0	2761	5	10	11	17	8	11	31	5	3	1	2	1	2	7	10
1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1+	2	0	3	0	0	0	0	0	0	0	0	0	0	0	1	0
1-	6	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0
2	4	0	0	0	8	0	1	0	0	0	0	0	0	0	0	0
2+	3	0	1	0	0	1	0	0	1	0	0	0	0	0	1	0
2-	7	0	0	1	0	0	8	0	0	1	0	0	0	0	0	0
3	11	0	0	0	0	0	0	6	0	0	0	0	0	0	1	0
3+	1	0	0	0	0	0	0	1	4	0	0	0	0	0	0	0
3-	2	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0
4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4+	1	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0
4-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
5+	5	0	1	0	0	0	0	0	0	0	0	0	0	0	6	0
5-	4	0	0	1	0	0	0	0	0	0	0	0	0	0	0	6

Figure 5: Confusion matrix of the agreement between annotator 1 and annotator 2 for label 2

D CONFUSION MATRICES PER ANNOTATOR PAIR

D.1 WESSEL & ANDREAS

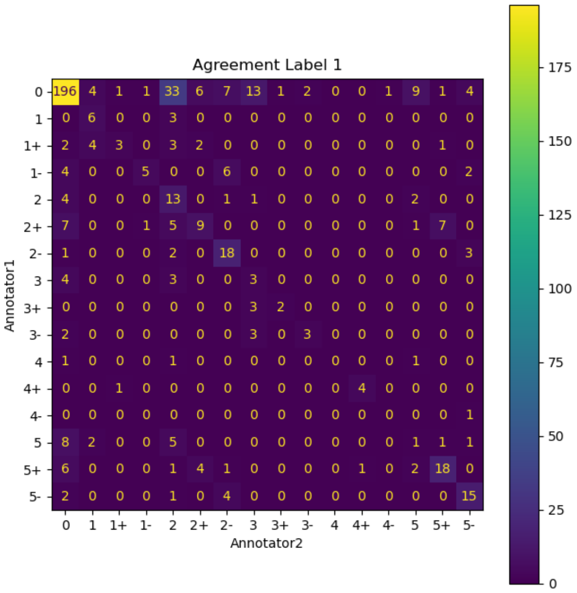


Figure 6: Confusion matrix for label 1

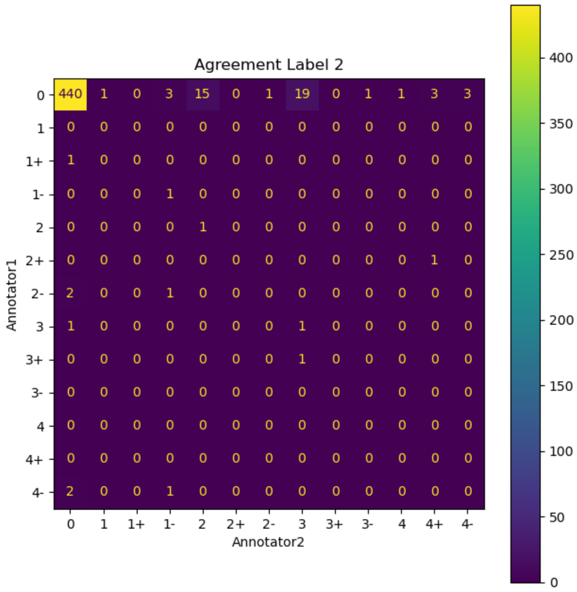
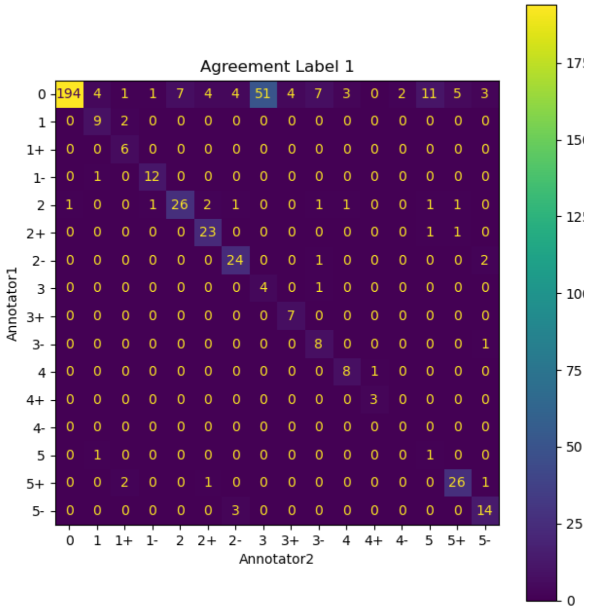


Figure 7: Confusion matrix for label 2

D.2 JANNICK & WESSEL



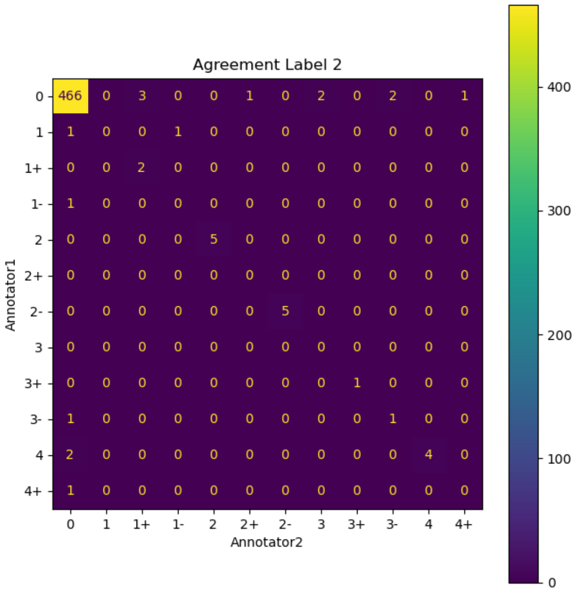
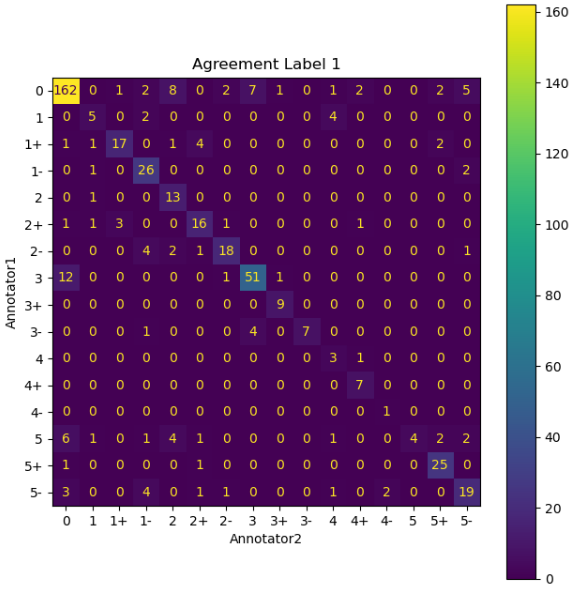


Figure 8: Confusion matrix for label 1

Figure 9: Confusion matrix for label 2

D.3 NIELS & JANNICK



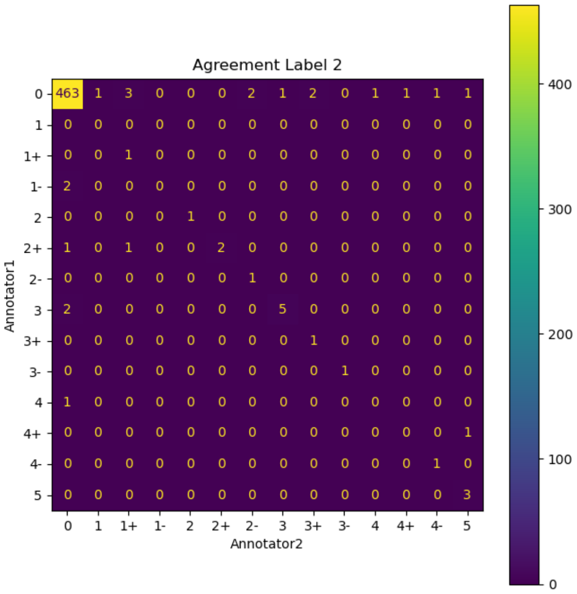
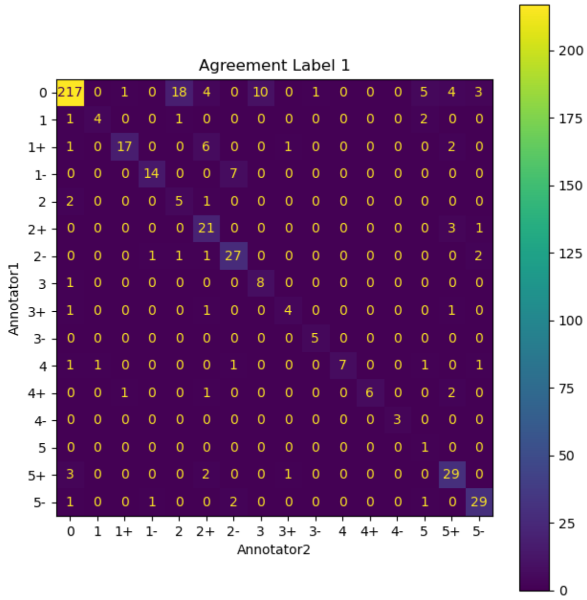


Figure 10: Confusion matrix for label 1

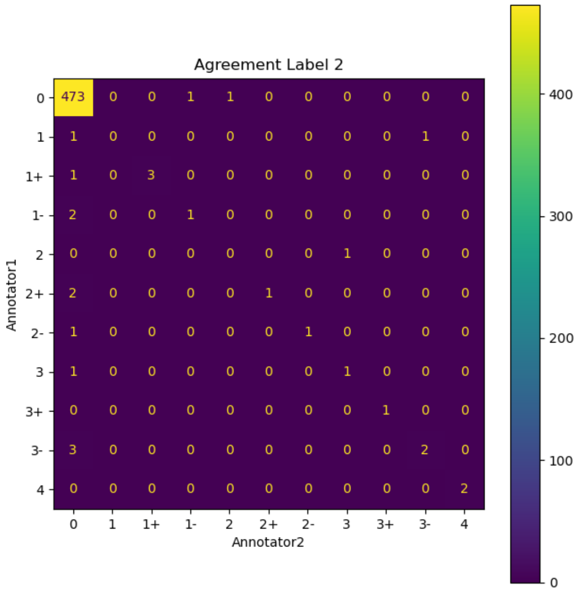
Figure 11: Confusion matrix for label 2

D.4 YOUNES & NIELS



Annotator1 \ Annotator2	0	1	1+	1-	2	2+	2-	3	3+	3-	4	4+	4-	5	5+	5-
0	217	0	1	0	18	4	0	10	0	1	0	0	0	5	4	3
1	1	4	0	0	1	0	0	0	0	0	0	0	0	2	0	0
1+	1	0	17	0	0	6	0	0	1	0	0	0	0	0	2	0
1-	0	0	0	14	0	0	7	0	0	0	0	0	0	0	0	0
2	2	0	0	0	5	1	0	0	0	0	0	0	0	0	0	0
2+	0	0	0	0	0	21	0	0	0	0	0	0	0	0	3	1
2-	0	0	0	1	1	1	27	0	0	0	0	0	0	0	0	2
3	1	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0
3+	1	0	0	0	0	1	0	0	4	0	0	0	0	0	1	0
3-	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0
4	1	1	0	0	0	0	1	0	0	0	7	0	0	1	0	1
4+	0	0	1	0	0	1	0	0	0	0	0	6	0	0	2	0
4-	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
5+	3	0	0	0	0	2	0	0	1	0	0	0	0	0	29	0
5-	1	0	0	1	0	0	2	0	0	0	0	0	0	1	0	29

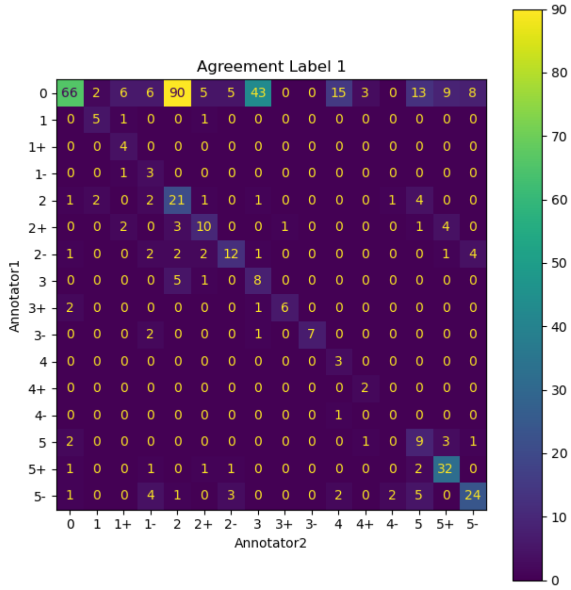
Figure 12: Confusion matrix for label 1



Annotator1 \ Annotator2	0	1	1+	1-	2	2+	2-	3	3+	3-	4
0	473	0	0	1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	1	0
1+	1	0	3	0	0	0	0	0	0	0	0
1-	2	0	0	1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	1	0	0	0
2+	2	0	0	0	0	1	0	0	0	0	0
2-	1	0	0	0	0	0	1	0	0	0	0
3	1	0	0	0	0	0	0	1	0	0	0
3+	0	0	0	0	0	0	0	0	1	0	0
3-	3	0	0	0	0	0	0	0	0	2	0
4	0	0	0	0	0	0	0	0	0	0	2

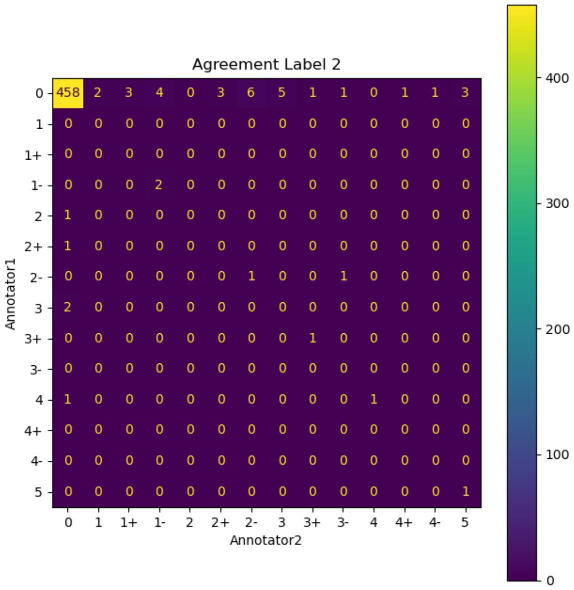
Figure 13: Confusion matrix for label 2

D.5 BORIS & YOUNES



Annotator1 \ Annotator2	0	1	1+	1-	2	2+	2-	3	3+	3-	4	4+	4-	5	5+	5-
0	66	2	6	6	90	5	5	43	0	0	15	3	0	13	9	8
1	0	5	1	0	0	1	0	0	0	0	0	0	0	0	0	0
1+	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0
1-	0	0	1	3	0	0	0	0	0	0	0	0	0	0	0	0
2	1	2	0	2	21	1	0	1	0	0	0	0	1	4	0	0
2+	0	0	2	0	3	10	0	0	1	0	0	0	0	1	4	0
2-	1	0	0	2	2	2	12	1	0	0	0	0	0	0	1	4
3	0	0	0	0	5	1	0	8	0	0	0	0	0	0	0	0
3+	2	0	0	0	0	0	0	1	6	0	0	0	0	0	0	0
3-	0	0	0	2	0	0	0	1	0	7	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0
4+	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0
4-	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
5	2	0	0	0	0	0	0	0	0	0	0	1	0	9	3	1
5+	1	0	0	1	0	1	1	0	0	0	0	0	0	2	32	0
5-	1	0	0	4	1	0	3	0	0	0	2	0	2	5	0	24

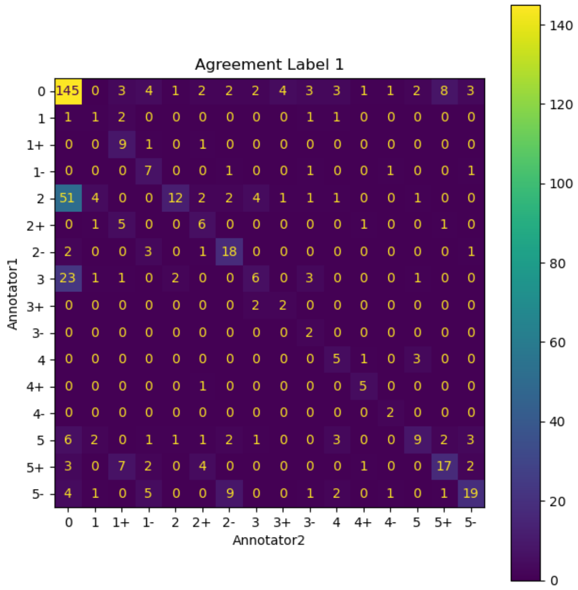
Figure 14: Confusion matrix for label 1



Annotator1 \ Annotator2	0	1	1+	1-	2	2+	2-	3	3+	3-	4	4+	4-	5
0	458	2	3	4	0	3	6	5	1	1	0	1	1	3
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1+	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1-	0	0	0	2	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0
2+	1	0	0	0	0	0	0	0	0	0	0	0	0	0
2-	0	0	0	0	0	0	1	0	0	1	0	0	0	0
3	2	0	0	0	0	0	0	0	0	0	0	0	0	0
3+	0	0	0	0	0	0	0	0	1	0	0	0	0	0
3-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	0	0	0	1	0	0	0
4+	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	1

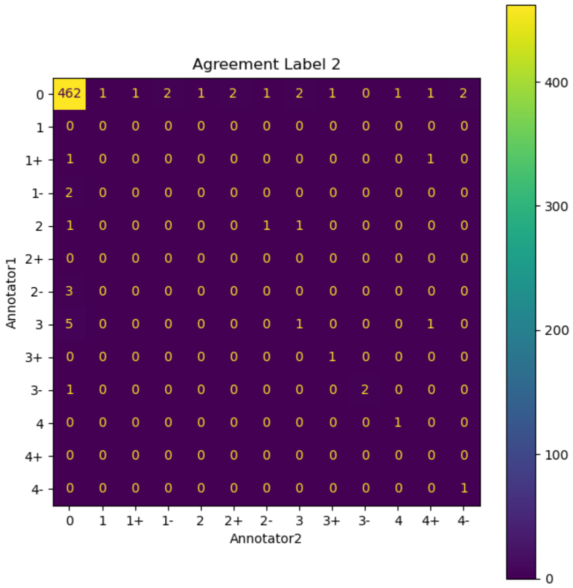
Figure 15: Confusion matrix for label 2

D.6 ANDREAS & BORIS



	0	1	1+	1-	2	2+	2-	3	3+	3-	4	4+	4-	5	5+	5-
0	145	0	3	4	1	2	2	2	4	3	3	1	1	2	8	3
1	1	1	2	0	0	0	0	0	0	1	1	0	0	0	0	0
1+	0	0	9	1	0	1	0	0	0	0	0	0	0	0	0	0
1-	0	0	0	7	0	0	1	0	0	1	0	0	1	0	0	1
2	51	4	0	0	12	2	2	4	1	1	1	0	0	1	0	0
2+	0	1	5	0	0	6	0	0	0	0	0	1	0	0	1	0
2-	2	0	0	3	0	1	18	0	0	0	0	0	0	0	0	1
3	23	1	1	0	2	0	0	6	0	3	0	0	0	1	0	0
3+	0	0	0	0	0	0	0	2	2	0	0	0	0	0	0	0
3-	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	5	1	0	3	0	0
4+	0	0	0	0	0	1	0	0	0	0	0	5	0	0	0	0
4-	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
5	6	2	0	1	1	1	2	1	0	0	3	0	0	9	2	3
5+	3	0	7	2	0	4	0	0	0	0	0	1	0	0	17	2
5-	4	1	0	5	0	0	9	0	0	1	2	0	1	0	1	19

Figure 16: Confusion matrix for label 1



	0	1	1+	1-	2	2+	2-	3	3+	3-	4	4+	4-
0	462	1	1	2	1	2	1	2	1	0	1	1	2
1	0	0	0	0	0	0	0	0	0	0	0	0	0
1+	1	0	0	0	0	0	0	0	0	0	0	1	0
1-	2	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	1	1	0	0	0	0	0
2+	0	0	0	0	0	0	0	0	0	0	0	0	0
2-	3	0	0	0	0	0	0	0	0	0	0	0	0
3	5	0	0	0	0	0	0	1	0	0	0	1	0
3+	0	0	0	0	0	0	0	0	1	0	0	0	0
3-	1	0	0	0	0	0	0	0	0	2	0	0	0
4	0	0	0	0	0	0	0	0	0	0	1	0	0
4+	0	0	0	0	0	0	0	0	0	0	0	0	0
4-	0	0	0	0	0	0	0	0	0	0	0	0	1

Figure 17: Confusion matrix for label 2