**6**

In this thesis, a text mining process in the field of chess annotations is introduced. An overview of the structure of a noted chess game and the different types of comments, NAG and symbols it contains is given. After preprocessing by NLTK, four different feature extraction models were created and compared using different classifiers, data sets and problems. The count-based model clearly performed best, followed by the TF-IDF-based model. Of the two word embeddings, the pre-trained Google News word embedding achieved the better accuracy values. Nevertheless, it could be shown that the word embedding learned on the chess comments also uses a meaningful assignment of vectors, whereby the similarities between words from chess vocabulary are even better and higher than in the Google News word embedding. Among the classifiers, there was no clear trend as to which would provide the best accuracies. A pattern was found when comparing the accuracies on the different data sets. Significantly higher accuracies could be achieved with the short comments than with mixed-length or long comments. This can be explained by the fact that, due to the initial filtering of comments for short comments, the proportion of English-language tokens is at least one third. \\\\

Overall, it can be stated that the achieved accuracies are not sufficient to make reliable classifications. Only for the two binary classification problems the accuracies are between 75\% and 80\%, which is still too error-prone to use these models for assigning annotation symbols to comments without annotation symbol yet. In the classification of the move comments into six classes, about 55\% accuracy was achieved, in the classification of the position comments into three classes about 60\% and in the classification of the position comments into seven classes only about 40\% accuracy. It also had to be noted that supposedly simple comments were misclassified in all models. The confusion matrices also indicated that for many instances only the most likely class was assigned because no criteria were found for a particular class. In the following, improvements in data selection, preprocessing, model generation and classification are presented that could solve these problems. \\\\

There is already a lot of room for improvement during the extraction process. A complex, but promising further development would be the parsing of the PGN files. This would make it possible to correctly recognize divided comments or comments preceding the annotation symbol. Conversely, comments incorrectly assigned to the preceding annotation symbol could be ignored. This would lead to a significant increase in data quality, since many cross-references to other games, which thus do not refer to the current move or position, would not be included in the data set. Alternatively, such statistical comments could also be filtered out by selecting a minimum proportion of English words instead of a minimum number of occurrences. An even bigger advantage achieved by parsing would be knowing which player made the last move. This would allow move comments, which describe the change on the board from an observer's point of view, to be transformed into consequences for the last player. Similarly, position comments could be better classified if the description from the point of view of the last player could be transformed to the consequences on the board for white and black. \\\\

In addition to the extraction process, optimizations can also be performed during preprocessing and feature selection. The removal of English stopwords turned out to be disadvantageous during the first tests, but a stopwords list created especially for chess games could bring an improvement. It could also be investigated whether lemmatizing is actually better than stemming or no normalization at all. Categorizations are also possible, e.g. all chess moves could be grouped into one category or the equivalent phrases "better is", "\#C4" and "$\$142$". Such measures lead to a quantitative deterioration, but a qualitative improvement of the features. Instead, besides the already included comment length, other interesting features could be added to the model, e.g. the final result of the chess game or the elo ratings of the chess players. In general, attributes of different models, e.g. count and TF-IDF values, can be combined with each other if no redundancies arise. \\\\

As mentioned above, the better results of the self-created word embedding based on the chess comments in comparison to the pretrained model via Google News are not reflected in the achieved accuracies in the classification. A possible reason has already been discussed with the problems of classification with terms that are too similar. An improvement could be achieved by learning word embedding on additional data such as comments without annotation symbols which of course could also lead to improvements in the count-based and TF-IDF-based model. Another possibility for improvement, which can be applied to both word embeddings, is the definition of a better function for calculating the comment vector. Especially with long comments many unimportant words can lead to the fact that the vectors of the decisive words hardly influence the comment vector. An interesting alternative model that uses tree structures to determine a final value for a comment from individual word sentiment values was developed by \citeauthor{Socher2013} \autocite{Socher2013}. \\\\

The classification process also offers some possibilities for further development. If larger computing capacities are available, an attribute selection only on the basis of the term frequencies could be avoided and be built into the classification process instead. Weka offers an AttributeSelectedClassifier, which selects a suitable subset of attributes only based on the training data. This procedure has already been applied to the preselected attributes, but could not lead to any further improvements. With the formation of ensembles from different classifiers, the accuracy values could also be increased. \\\\

Finally, the target classes must also be questioned. A strict subdivision into move and position comments is questionable; in this case a multilabel classification might be conceivable. For the move and position comments it would have to be checked whether a fine partition into six or seven classes is possible at all or whether the additional classes overlap too much. Otherwise the reduction to two or three classes would remain, or the adjustment of the costs to a very low value, so that a choice of the "neighbor" class would be punished only slightly. In addition to the final evaluation, these cost values can also be included in the learning process. Weka offers the CostSensitiveClassifier for this purpose, which, however, did not lead to any significant improvement in costs during initial tests.