Comparison of a TF-IDF and Word Embedding Approach to Develop an Automated Hate Speech Classifier for Tweets

Joshua Mähl
B. Sc. Student
Business Administration
and Computer Science
Nordakademie

Tim Zielke
B. Sc. Student
Business Administration
and Computer Science
Nordakademie

{joshua.maehl, tim.zielke}@nordakademie.de

Abstract

In this study, we examine the classification of tweets as Hate Speech with the help of model-driven Artificial Intelligence. We use two different approaches: TF-IDF and Word Embedding. We normalise the underlying data through pre-processing and we describe the procedure and background of both approaches. We cannot recommend one approach unconditionally for every possible situation, as our results reveal that both approaches are suitable for different use cases. The use of TF-IDF corresponds to a stricter Hate Speech policy for social media platforms and Word Embedding is a more efficient choice for a relative laissez-faire policy.

1 Introduction

Over the past decades, technology has been developing at a rapid pace, being normalized not only in research areas but also in business and private life. It is generally known that our lives have been changed by the rise of technologies, such as the Internet or mobile devices, and will probably continue to change in the future.

Social media can be classified in this context. Companies such as Facebook, Twitter and Google are becoming part of everyday life and are gradually merging with the habits of billions of people who use the offered services. The world is becoming smaller and more connected through instant communication. This "new" form of communication offers, on the one hand, a digital protective wall to the recipient and, on the other hand, it provides a platform for all users to exercise the human right of freedom of expression.

However, a negative side effect is a potential abuse through Hate Speech, as groups of people can be defamed very easily. Consequently, social network companies might have the responsibility to control their content by filtering out offensive posts.

Due to the immense amount of data, it is impossible to check the contents manually. Furthermore, the field of Artificial Intelligence offers promising possibilities to solve this problem algorithmically. Some research in this field is based on the application of pre-trained Word Embeddings, which can represent even colloquial words as vectors. Besides, there are other forms of natural language representation for machine learning algorithms, such as the TF-IDF.

However, our literature research showed that there is a scarcity of research on the comparison of different approaches, as individual methods are generally studied in detail. Therefore, the focus of this paper will be on discussing advantages and disadvantages, as well as similarities and differences between the approaches described above. We test the following hypothesis: *A Hate Speech Detection experiment based on TF-IDF as a stand-alone feature outperforms Word Embedding experiments.*

We analyse topic-related and peer-reviewed research and examine the given, classified training data set of about 900 tweets using word clouds. In addition, we perform pre-processing on the complete data set, whereby a normalized and cleaned form of the tweets is achieved. This form is the input for Word Embedding on the one hand

and the vector representation via TF-IDF on the other hand. After vectorization, different classification algorithms are used: A Naive Bayes Classifier for the TF-IDF approach and the Ada-Boost algorithm for Word Embedding. After visualizing our results, we discuss the reasons behind our findings and compare both approaches.

Our key findings are as follows:

- No approach completely outperforms the other one. Depending on a perspective, an approach can be recommended.
- TF-IDF offers 89% Accuracy and 88% F1-Value.
- Word Embedding offers 87% Accuracy and 87% F1-Value.

2 Related Work

Fortuna and Nunes carried out a study to investigate the state-of-the-art in Hate Speech detection. Firstly, they describe the fight against hateful messages in social media as a technological and social problem that is receiving increasing political attention (e.g. European Union Commission pressuring social media platforms to renew their policies) [1, p. 85:2]. They also count studies, which have been published on this topic since 2004 and found exponential growth between 2013 and 2017 [1, p. 85:11]. The authors observe that "comparative studies and surveys are also scarce in the area" [1, p. 85:22], which further motivates the focus of our experiment.

Mondal et al. conducted a study to measure Hate Speech on social media platforms. They define Hate Speech as "an offensive post, motivated, in whole or in a part, by the writer's bias against an aspect of a group of people" [2, p. 87] and propose a categorisation of Hate Speech target groups: race, behaviour, physical, sexual orientation, class, gender, ethnicity, disability, religion [2, p. 89]. Out of 512 million tweets, their model identified 20,305 Hate Speech posts, 86% of which were addressed to individuals belonging to the groups "race" and "behaviour" [2, pp. 88-89].

Lilleberg and Yun Zhu conducted different experiments to investigate the potential of vector representation by Word2Vec for text classification. Word2Vec is a Word Embedding variant that provides a vector representation of each word within a document due to its context and semantic characteristics and "brings extra sematic [sic] features that help in text classification" [3, p. 136]. According to the authors, this approach is compliant with common text classification techniques like Naive Bayes classifier, Support Vector Machines and TF-IDF [3, p. 138]. They conclude that a combination of TF-IDF and Word2Vec does not generally outperform an individual application of both approaches [3, p. 136].

Interestingly, there seems to be dissent about the applicability of Word Embeddings. On the one hand, Li et al. categorise Word Embeddings as "effective in many natural language processing and text modelling tasks" [4, p. 651], while Schmidt and Wiegand attest this method only limited effectiveness, even if their approach uses pre-trained embeddings for Hate Speech Detection [5, p. 3].

MacAvaney et al. conducted another study in the field of Hate Speech detection. The authors use a multi-view Support Vector Machine approach using TF-IDF as a feature. They note that this approach calculates the relative importance of a word and is based on the bag-of-words assumption, where words are considered without grammar or specific order [6, p. 7]. The model includes n-grams, that is in their case, phrases of n words with 1 < n < 5. They conclude that the unigram model gives the most accurate results, although each model examines a different aspect of Hate Speech [6, p. 11].

Another important aspect of our topic is examined by Xia et al. They investigate reasons why some state-of-the-art detection models have a high False Positive rate. It is a problem for social media platforms when this rate is up to 46% and therefore "innocent" statements are misclassified as Hate Speech [7, p. 7]. The authors ascribe this to the existence of a racial bias, which can be mitigated by demoting this attribute [7, p. 12].

On the other hand, Davidson et al. attribute the False Positive problem to the use of bag-of-words approaches because "the presence of offensive words can lead to the misclassification of tweets as hate speech" [8, p. 1]. The authors argue that TF-IDF models can achieve a high Recall, but that the legitimacy of the model must be considered individually and might not be given [8, p. 1]. Their key finding, consequently, is that tweets without explicit hate keywords are more difficult to classify as Hate Speech [8, p. 1].

Wiegand et al. continue this idea by pointing out the existence of biased datasets. The reason might be that researchers perform focus sampling instead of random sampling. Consequently, the dataset should have a realistic proportion of Hate Speech [9, p. 602]. Additionally, Waseem explains unreal data through the background of the annotator of the Hate Speech classification. The author researched that amateur annotators have a lower inhibition threshold for identifying Hate Speech than expert annotators [10, p. 138].

3 Data

Before we start the actual experiment, it is important to obtain an overview of the data. Our training and test data were retrieved from Davidson et al. [8]. We will examine a training data set of 902 tweets with corresponding classification (1 means Hate Speech and 0 Non Hate Speech). We also validate our model using a test data set of 100 classified tweets. The train-test-split is consequently 90% to 10%.

Due to the relatively small amount of data, an experiment using Deep Learning is not very promising, because this is only the most efficient learning method when a large amount of data is available. The data provides a good fundament for an analysis using "classical" Machine Learning methods.

In order to better understand the data, we separated Hate Speech tweets and normal tweets and then created a visualisation using the wordcloud library from Python 3 for each classification, which was done after the pre-processing of the tweets to clarify the essence of the tweets. In the following, the results of the visualisation are shown:

Figure 1: Word Cloud representing common expressions in Hate Speech tweets

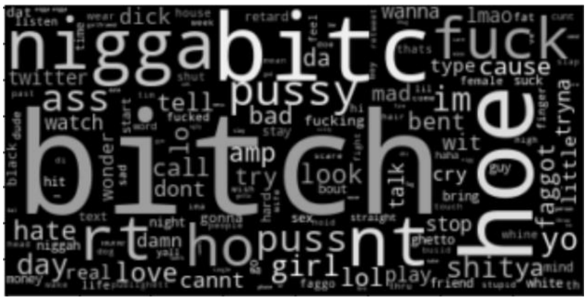
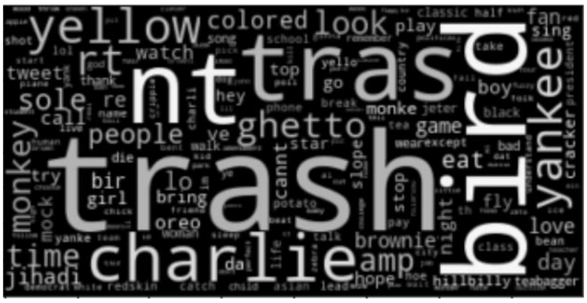


Figure 2: Word Cloud representing common expressions in Non Hate Speech tweets



The figures clearly show that a different vocabulary is used when writing Hate Speech tweets. Words (or common derivations of the word) such as "bitch", "nigga", "hoe", "pussy" etc. are the most commonly used terms (see Figure 1). A distinctive peculiarity of many negative tweets is the widespread use of different syntactical compositions, such as "Nigga" and "B*tch" instead of "Nigger" and "Bitch". Therefore, matching with a static Bad Word List might not be an effective strategy for Hate Speech Detection if it does not include derivatives. On the other hand, there is no recognisable thematic pattern in other tweets (see Figure 2). This was expectable due to the relatively large semantic variety of "normal" tweets.

Another special characteristic of the present data is that half of the tweets are classified as Hate Speech. However, this is not the case for a real sample of tweets. In general, the proportion of Hate Speech tweets is significantly lower, as some before stated research (see Mondal et al. [2]) has already shown. In order to reveal general characteristics of Hate Speech, it is nevertheless advisable to ensure that the data is divided equally. Thus, the model might be independent of the circumstance of unequal distribution.

4 Methodology

4.1 Pre-processing

Our first step in applying the algorithms consisted of cleaning the data by using pre-processing. For this, we chose the Data Mining tool KNIME Analytics Platform. First, all characters were set to lowercase, because upper and lowercase are not relevant for our approaches and because we want to use a standardised cleaned database as input for the algorithms. In the following steps, individual tweet components were removed, which also do not provide any semantic meaning. These include the following components: References to user names (e.g. @user123), the term rt for Retweet and all hyperlinks (http and https). We used regular expressions to automatically identify these components.

In the following steps, we removed all punctuation signs, also for the same reason as switching all characters to lowercase. Furthermore, a list of stop words was compared with the database and each stop word was removed, as they provide little semantic value. This is a built-in list in KNIME, which is offered in a standardised form for all Natural Language Processing projects.

Besides, we carried out a lemmatizing, which reduced various word forms to their basic form (the lemma). We chose to lemmatize instead of stemming because it is based on a natural language dictionary and therefore not only recognises the root of a word but can also relate irregular word forms to the lemma (e.g. ran becomes run). In this way, we achieve a lower word diversity, which is particularly advantageous for vectorisation, as similarities are abstracted. KNIME offers an implementation of a Stanford Lemmatizer, which we used for our data.

4.2 TF-IDF Approach

Our first approach consists of vectorising individual words and documents using the Term Frequency - Inverse Document Frequency (TF-IDF) method. In our project, the respective tweets are regarded as documents. TF-IDF is a useful feature for making decisions, as Wu et al. concluded in their study [11, p. 13:1]. According to the conclusion made by MacAvaney et al. [6, p. 11] (see Related Work), we use unigrams for our experiment.

Therefore, we have created a Python script that reads and processes the pre-processed data. For the present approach, we use the library sklearn. On the one hand, we import the classes TfidfVectorizer and CountVectorizer for vector conversion of the CSV files. Furthermore, we use the class MultinomialNB from the same library to train our model using the Naive Bayes Classifier.

First, an object of the class TfidfVectorizer is instantiated. On this object, we run the function *fit_transform* and pass the tweets of our training set to the function. This function serves on the one hand to calculate the TF-IDF of the documents and on the other hand to transform them into a vector. The transformed data is stored in a variable buffer to train the model with this data later on. Analogously, we calculate the TF-IDF and create the vectors of the test set.

To implement the Naive Bayes classification we create an object of the class MultinomialNB. We train the model by running the function *fit* and using the training vectors as X-value and the corresponding Hate Speech classification as Y-value. We then validate this model by executing the *predict* function, which takes the vectors of the test set (X-value) as input and tries to predict the corresponding Y-value. The predicted and the actual Y-values are compared and in this way, we calculate the classification metrics. Our source code can be retrieved from this[1] repository.

We consider TF-IDF to be a useful classification feature. The idea relies on the assumption that by removing the irrelevant but frequently used stop words, only relevant terms are left. Based on the importance of a word, it can be concluded relatively easily whether it is Hate Speech. The algorithm handles the weighting of the individual words and is scientifically established and reliable since it has been used in many topic-related research projects (see e.g. MacAvaney et al. [6]).

We decided to use the Naive Bayes Classifier because of its ability to develop a well functioning model with a relatively small amount of training data. Compared to other studies, which had several thousand (and sometimes millions) of available and classified tweets, our training set comprises 900 classified tweets.

[1] https://github.com/JoshuaMaehl/hate-speech-classifier

4.3 Word Embedding Approach

Our second approach involves the use of Word Embedding. These are pre-trained databases that convert a word in its context into a high-dimensional vector. Again, several methods can be used in state-of-the-art solutions. These include Word2Vec and FastText.

Some researchers have trained models based on particular information bases in the past. For example, there are Word Embeddings that are specialized for social media applications. For this reason, we use a Word2Vec model that has been trained with a very large amount of Twitter data and was created and firstly presented by Godin et al. in 2015 in their publication about recognition of named entities in Twitter [12]. The corresponding GitHub repository from which we retrieved the file and information can be found here[2].

As well in this approach, we created a Python script that reads and processes the CSV files. For this, we imported the library gensim. This library is necessary to read and import a pre-trained Word Embedding model using the function *load_word2vec_format*. We also use pandas to create Data Frames in which the vectors are added gradually.

Similar to the other approach, we read the pre-processed training and test sets as CSV files, yet without splitting them up. The tweets are then split into words and each word is vectorised according to its defined 400-dimensional weighting. As stated above, all words are appended gradually to a data frame, which represents a tweet. The document vector is the arithmetic mean of the word vectors. In a for-loop construct, we append each tweet vector to the entire training (or test) data frame.

The classifications (Hate Speech or not) are added to the data frame. Then we divide the vectorised data into x and y values: train_x, train_y, test_x and test_y. This is necessary to train the classification algorithm.

We wanted to implement both approaches with the same classifier (Naive Bayes). Unfortunately, the Naive Bayes algorithm cannot process negative input data, so we import the Ada Boost Classifier as a substitute. We use the same algorithm parameters (e.g. learning rate is 0.1) for better comparability. Similar to the first approach, we predicted the test_y values and compare the output with the actual values to calculate the classification metrics.

As stated in the Related Work Section, Word Embeddings offer a valid basis for the analysis and classification of Hate Speech tweets. Lilleberg and Yun Zhu even combined both TF-IDF and Word Embeddings feature sets [3]. Consequently, we consider it a necessity to compare the two approaches. We also found the Word2Vec model presented by Godin et al. [12] to be an optimal tool, which was trained on Twitter data.

In our work, we use the AdaBoost Classifier. This is an algorithm often applied to binary classifications and, according to Rajesh and Dhuli, the boosting algorithm is a robust learner that derives from a weighted combination of weak learners [13, p. 247]. The weighting is done with a value so that the error rate in each weak learner is minimal [13, p. 247].

5 Results

Respecting the information from previous Sections, we conducted our experiment following literally every description. In the following, a systematic overview of our results is shown:

Table 1: Confusion matrix for Hate Speech classification based on TF-IDF feature

Table 2: Confusion matrix for Hate Speech classification based on Word2Vec features

	Actually Hate Speech	Actually Not Hate Speech
Predicted Hate Speech	41	8
Predicted Not Hate Speech	3	48

	Actually Hate Speech	Actually Not Hate Speech
Predicted Hate Speech	44	4
Predicted Not Hate Speech	9	42

[2] https://github.com/FredericGodin/TwitterEmbeddings

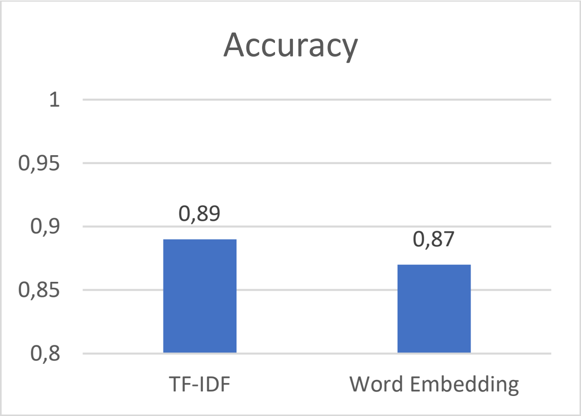


Figure 3: Comparision of Accuracy from both approaches

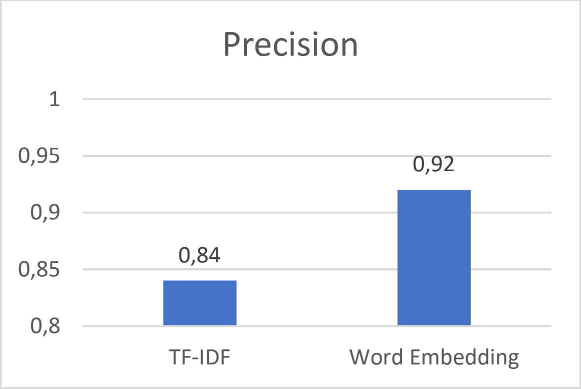


Figure 4: Comparision of Precision from both approaches

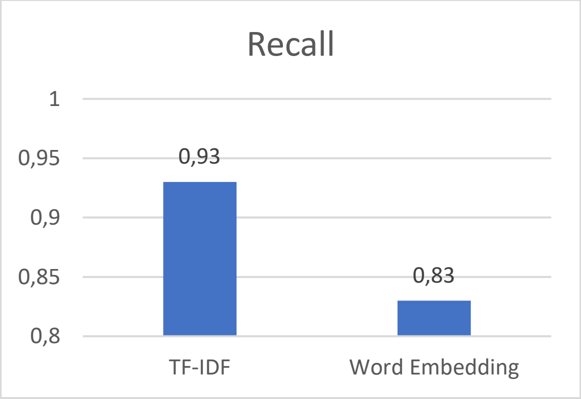


Figure 5: Comparision of Recall from both approaches

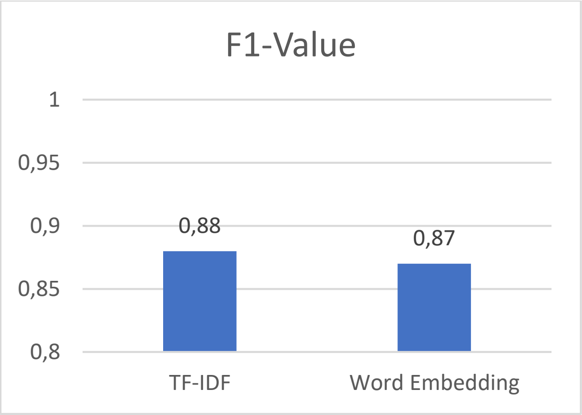


Figure 6: Comparision of F1-Value from both approaches

6 Discussion

The findings of this study clearly show that the TF-IDF has slightly higher Accuracy than the compared Word Embedding approach. This corresponds to a value of 0.89 as compared to 0.87 (see Figure 3). These values can be calculated directly from the Confusion Matrix: 89 out of 100 tweets are either True Positive or True Negative, while there are 8 False Positive and 3 False Negative values when using TF-IDF as a feature (see Table 1).

On the other hand, our other model achieves 86 correctly classified tweets and 13 incorrectly classified tweets when using Word2Vec (see Table 2). Therefore, our approach was only able to process 99 out of 100 test tweets. This leads us to believe that with this model, there is a possibility that individual tweets could not be processed correctly (possibly caused by errors during vectorisation).

Accuracy is a useful but not always reliable classification measure because it may not accurately describe the reality. For example, if 100 out of 10,000 tweets were Hate Speech tweets (which is a valid assumption based on our literature research) and the model classified all tweets as Non Hate Speech, then Accuracy would be almost perfect at 99%. However, this would result in the model not fulfilling its purpose.

Therefore, we calculate other metrics to evaluate our two models: Precision, Recall and F1-Value. TF-IDF has a lower Precision (0.84 compared to 0.92, see Figure 4), while it provides a higher Recall value (0.93 compared to 0.83, see Figure 5). Consequently, the F1-Value for our TF-IDF approach is slightly higher (0.88) than for our Word Embedding approach (0.87) (see Figure 6). This leads us to another assumption: the Accuracy value has the expected significance in our experiment.

The reason why TF-IDF worked well could be related to the fact that the frequency of terms occurring within a document and the distribution of the terms throughout the text corpus is used to prioritise the terms internally, which, in conjunction with the classification already given, can be used to decide whether a tweet is classified as Hate Speech. This finding confirms for our experiment the statement of Davidson et al. [8] that a text classification with TF-IDF can achieve a high Recall (see Related Work).

It is also not surprising that the result of the classification using Word2Vec is useful. This could be related to the Word Embedding model used by Godin et al. [12], which was trained exclusively on Twitter data. Consequently, the model already "knows" certain connotations of what constitutes Hate Speech. The strength of the Word2Vec algorithm lies in the high-dimensional vectorisation of individual words and documents. This allows the model to be mainly correctly classified, although we had a relatively small amount of data available.

After considering the Accuracy, it is also important to analyse the other classification metrics for the evaluation of the experiment. For this purpose, we calculated the values Precision and Recall and put them in relation by calculating the F1-Value.

Our result says that the use of Word Embeddings results in a higher Precision. This means that the proportion of correctly positive classified Hate Speech Tweets (True Positive) from all as Hate Speech classified Tweets (Predicted HS) is higher than using the compared approach.

Furthermore, our TF-IDF model achieves a significantly higher Recall value. This means that the proportion of correctly positive classified Hate Speech Tweets (True Positive) from all actual Hate Speech Tweets (Actual HS) is higher than using the compared approach.

Finally, we consider the F1-Value. This attribute should be seen as a summarising metric, being the harmonic mean between Recall and Precision. From the previous paragraphs, it can be concluded that both approaches had a significantly higher value for one metric each, which balances each other out when calculating the F1-Value. Consequently, we get a slightly higher value of 0.88 for TF-IDF and 0.87 when using Word2Vec (see Figure 6).

In addition to the effects of the calculated results, we analyse the presumed reasons. First, we look at the background of the TF-IDF experiment. Apparently, our model is able to support the assertion of Davidson et al. that this feature can achieve a high Recall value [8, p. 1] (see Related Work). In fact, the value significantly exceeds the compared approach by 10%.

The high Recall value can be attributed to the bag-of-word approach of the underlying experiment. Outside its context, a word can be identified as offensive through its "importance" in the document and the text corpus. This reduces the probability of classification as a False Negative, as the model learns about the patterns of different structures of Hate Speech tweets using this feature. On the other hand, ignoring a word's context offers a possible explanation for the low Precision value of the TF-IDF model.

Finally, during our data review in Section 3 we found that many tweets contain offensive words, but in fact, do not contain a negative statement on a semantic level (e.g. "Life can be a bitch"). This increases the probability that "innocent" tweets are found to contain Hate Speech and increases the False Positive rate. This pattern can directly be deduced from our experiment.

Otherwise, there are some possible explanations for the outcome of the use of Word Embeddings. As presented in Section 4, we use a pre-trained model by Godin et al [12]. Its suitability for this purpose is beyond doubt, as confirmed by our model and several related pieces of research. However, there are significant discrepancies between Recall (0.83) and Precision (0.92).

The high Precision value exceeds the compared approach significantly by 8%. Similar to the TF-IDF approach, this can be attributed to a relatively high number (9 out of 100) of False Negatives. One explanation for the nondetection of Hate Speech could be the largely unknown origin and development of the pre-trained model. Some authors, such as Xia et al. [7], Wiegand et al. [9] and Waseem [10], recognise that biased data sets might cause a model to have some preliminary assumptions that affect the outcome. We do not have any information about other studies examining a potential bias for Godin's Twitter model [12]. We cannot therefore rule out the possibility that certain terms or topics were incorrectly trained as Non Hate Speech. The definition is a subjective assessment; we base our opinion on the definition by Mondal et al. [2, p. 87] and the presented target groups (see Related Work).

Another possible explanation for the result of our World Embedding model could be the time-lapse since the publication of Godin et al.'s pre-trained Twitter Embedding [12]. Since 2015, many aspects of communication, including insults, have changed in the English-speaking cultural area, as it is a normal evolution of natural language. Hence, we assume that the model might misjudge some of the relevant topics in 2020 and fail to recognise certain insults (e.g. related to the Coronavirus).

Wiegand et al. explain that focus sampling may lead to a distortion of the model [9, p. 602], that is, if a large number of Hate Speech tweets are used to create the model. In fact, our training data does not show a realistic split between Hate Speech and Non Hate Speech (less than 1% of a random and large amount of data could be offensive), but the split is half. Similarly, there is no information about the creation of the Twitter Embedding model which Waseem points out as the unknown annotator problem [10] (see Related Work). Consequently, we cannot rule out the possibility that this may have an impact on the validity of our experiment despite our promising results.

We also question what would have improved our experiments. A general suggestion is to look at additional metadata, such as demographic data about the users. Although the Twitter API does not provide direct information for privacy reasons, there are alternatives. Waseem and Hovy include information on gender and geographic origin in their Hate Speech detection. A higher F1-Value is achieved when gender is also considered [14, p. 92]. Using the same classifier algorithm for both approaches might also improve validity.

As a conclusion of our discussion, both approaches can now be compared. For this purpose, we use the summarising F1-Value. At 88%, TF-IDF achieves a slightly higher value than Word Embedding (87%) (see Figure 6). Nevertheless, this finding does not allow us to recommend the first approach without exception. Based on the discrepancies between Recall and Precision, we recommend the following: If a social media platform aims to achieve a low False Positive rate, which means reducing the number of messages that are classified as Hate Speech without containing offensive content, Word2Vec provides more efficient features. However, if the highest possible coverage of actual Hate Speech is desired, then the use of TF-IDF should rather be considered.

This leads us back to the hypothesis formulated in the introduction. After considering our results, we must reject this hypothesis. No clear preference for an experiment can be concluded because the recommended choice depends on the factors described in the previous paragraph.

7 Conclusion

This experiment aimed to create two models capable of classifying tweets according to the presence of Hate Speech. To do this, we first analysed the data and decided to clean it through pre-processing and to bring it into a normalised form. On this basis, two models could be developed: one based on TF-IDF as a feature and one using Word2Vec for vectorisation.

We conclude that the choice of model depends on the desired purpose. If the purpose is to cover as much actual Hate Speech as possible and accept the possible misrecognition of Non Hate Speech, then TF-IDF is the better choice. This would correspond to a stricter Hate Speech policy on a social media platform. On the other hand, the aim might also be to guarantee users as much freedom as possible, in other words, to accept that real Hate Speech will eventually not be recognised. Word Embedding is a more efficient method for this situation.

There is some optimisation potential and limitation for our models. One point that we believe could increase efficiency is the amount of data because related studies use significantly more tweets. Furthermore, the same split between Hate Speech and Non Hate Speech is unlikely to be true in reality. Consequently, we recommend a larger amount of data with a more realistic distribution. An examination of the bias of the underlying data is also recommended for future pieces of research. Besides, our study was limited to English tweets, which means that the application of our results in other languages cannot be guaranteed generally without further examinations.

References

[1] P. Fortuna and S. Nunes. "A Survey on Automatic Detection of Hate Speech in Text," *ACM Computing Surveys*, Vol. 51, No. 4, Article 85, pp. 85:1 – 85:30, July 2018. [Online]. Available: DOI: https://doi.org/10.1145/3232676 [Accessed Nov. 27, 2020].

[2] M. Mondal *et al.*, "A Measurement Study of Hate Speech in Social Media," In *Proceedings of HT '17: 28th Conference on Hypertext and Social Media*, July 04-07, 2017, Prague, Czech Republic, pp. 85 – 94. [Online]. Available: DOI: http://dx.doi.org/10.1145/3078714.3078723 [Accessed Nov. 28, 2020].

[3] J. Lilleberg and Y. Yun Zhu, "Support vector machines and Word2vec for text classification with semantic features," In *2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC)*, July 06-08, 2015, Beijing, China, pp. 136 – 140. [Online]. Available: DOI: 10.1109/ICCI-CC.2015.7259377 [Accessed Nov. 28, 2020].

[4] S. Li *et al.*, "Adaptive Probabilistic Word Embedding," In *Proceedings of The Web Conference 2020 (WWW'20)*, April 20–24, 2020, Taipei, Taiwan, pp. 651 – 661. [Online]. Available: DOI: https://doi.org/10.1145/3366423.3380147 [Accessed Nov. 28, 2020].

[5] A. Schmidt and M. Wiegand, "A Survey on Hate Speech Detection using Natural Language Processing,", In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, April 3-7, 2017, Valencia, Spain, pp. 1 – 10. [Online]. Available: DOI: 10.18653/v1/W17-1101 [Accessed Nov. 26, 2020].

[6] S. MacAvaney *et al.*, "Hate speech detection: Challenges and solutions," *PLoS ONE,* Vol. 14, No. 8, pp. 1 – 16, August 2019. [Online]. Available: DOI: https://doi.org/10.1371/journal.pone.0221152 [Accessed Nov. 28, 2020].

[7] M. Xia *et al.*, "Demoting Racial Bias in Hate Speech Detection," In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, July 10, 2020, Online, pp. 7 – 14. [Online]. Available: DOI: 10.18653/v1/2020.socialnlp-1.2 [Accessed Nov. 25, 2020].

[8] T. Davidson *et al.*, "Automated Hate Speech Detection and the Problem of Offensive Language," In *Proceedings of the Eleventh International AAAI Conference on Web and Social Media*, May 15–18, 2017, Montréal, Canada. Palo Alto: The AAAI Press, 2017.

[9] M. Wiegand *et al.*, "Detection of Abusive Language: the Problem of Biased Datasets," In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, June 2 - 7, 2019, Minneapolis, USA, pp. 602 – 608. [Online]. Available: DOI: 10.18653/v1/N19-1060 [Accessed Nov. 28, 2020].

[10] Z. Waseem, "Are You a Racist or Am I Seeing Things? Annotator Influence on Hate Speech Detection on Twitter," In *Proceedings of 2016 EMNLP Workshop on Natural Language Processing and Computational Social Science*, November 5, 2016, Austin, USA, pp. 138 – 142 [Online]. Available: DOI: 10.18653/v1/W16-5618 [Accessed Nov. 30, 2020].

[11] H. Wu *et al.*, "Interpreting TF-IDF term weights as making relevance decisions," *ACM Transactions on Information Systems*, Vol. 26, No. 3, Article 13 pp. 13:1 – 13:37, June 2008. [Online]. Available: DOI: http://doi.acm.org/10.1145/1361684.1361686 [Accessed Dec. 01, 2020].

[12] F. Godin *et al.*, "Multimedia Lab @ ACLW-NUT NER Shared Task: Named Entity Recognition for Twitter Microposts using Distributed Word Representations," In *Proceedings of the ACL 2015 Workshop on Noisy User-generated Text*, July 31, 2015, Beijing, China, pp. 146 – 153. [Online]. Available: DOI: 10.18653/v1/W15-4322 [Accessed Dec. 02, 2020].

[13] K. Rajesh and R. Dhuli, "Classification of imbalanced ECG beats using re-sampling techniques and AdaBoost ensemble classifier," *Biomedical Signal Processing and Control*, Vol. 41, pp. 242 – 254, March 2018. [Online]. Available: DOI: https://doi.org/10.1016/j.bspc.2017.12.004 [Accessed Dec. 02, 2020].

[14] Z. Waseem and D. Hovy, “Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter,” In *Proceedings of the NAACL Student Research Workshop*, June 13 – 15, 2016, San Diego, USA, pp. 88 – 93. [Online]. Available: DOI: 10.18653/v1/N16-2013 [Accessed Dec. 03, 2020].