First Year Project - Project 3

Natural Language Processing

Group 8

Ida Maria Zachariassen
idza@itu.dk

Sabrina Fonseca Pereira
sabf@itu.dk

Magnus Sverdrup
magsv@itu.dk

Rasmus Bondo Hansen
rabh@itu.dk

Ruben Oliver Jonsman
rubj@itu.dk

June 3, 2021

Contents

1 Introduction

Artificial intelligence has become part of our everyday lives – virtual assistants, autocorrect, chatbots and translators. They all use machine learning algorithms and Natural Language Processing (NLP) to process and respond to human language. Every day a staggering amount of unstructured text data is generated. From medical records to social media tweets, automation is critically needed to analyze every aspect efficiently and correctly.

This project explores how machine learning models, trained on Twitter data, perform when trying to classify the writer's intention. The focus was split into two categories, one predicting irony and the other predicting stance regarding atheism. Classifying intention in social media is a powerful tool. It can be used for business purposes, such as gauging how a brand is perceived by its customers, critical applications such as public safety by detecting criminal intent and moderating hate speech.

Human language is complex and diverse, which makes it very challenging for machines to process and understand. For this reason there were tested different machine learning models and developed different tokenizers to find out what combination yielded the best performance when classifying tweets.

2 Data and Preprocessing

The datasets used in this research are part of the TweetEval corpus, a collection of 7 datasets for different classification tasks. Each task had a test, train and validation file, consisting of one tweet per line with a corresponding label file. Labels were numeric and mappings were provided with the corresponding meanings. For this project it was chosen to work with the irony and stance datasets, and for stance, the focus was on atheism.

During pre-processing, the data was checked for equality between the number of tweets and labels for all of the used datasets. The proportion between development and testing data for both tasks was also checked, showing roughly 19% testing data available for irony dataset, and 30% testing data for the atheism set.

To further explore the performance of the model, all stance datasets (abortion, atheism, climate, feminist and hillary) were merged into one set. The labels were re-mapped giving three unique stances to each topic. There was a big imbalance between the different classifications, indicating future problems for predicting the correct labels.

Tokenizers. To create the input for the machine learning models, a few different tokenizers were created. As a starting point they were aimed at segmenting the lines at normal "words". The tokenizers would then differ in their way of capturing the significance of written language on social media platforms like Twitter. An "ideal" tokenizers was created, which was set to capture only words, numbers, special characters (!?...&) and making everything lower case. Other tokenizers were made to capture only the hashtagged words, capture only non-words and capture only emojis. To ensure consistency and quality of the tokenizers, an evaluation between the baseline tweet tokenizer from the NLTK library and the "ideal" tokenizer was performed. A small subset of the training data, which had been taken out from the beginning, was used to quantitatively and qualitatively compare the outputs. Quantitatively, the "ideal" tokenizer and the baseline tokenizer matched on average 65%.

2.1 Data statistics

The binary classification model was trained on 2862 tweets from the irony dataset, which were almost perfectly balanced between ironic and not ironic tweets. The multi classification model was trained on 461 tweets from

the atheism dataset and were unevenly balanced between the three stances.

Train dataset	Tweets	Words	Characters
Irony	2862	38939	226812
Atheism	461	8388	49704

Table 1: Basic statistics on training datasets

When comparing the irony and atheism datasets, the main observed difference was their balance and the amount of data. As shown in Table 2 and Table 3, the irony training dataset has almost perfect balance, unlike the atheism training dataset, where a big imbalance between the classes was seen. This imbalance can become an issue as the model would be biased towards 'against' and to 'none', to lesser degree, when predicting stances in atheism.

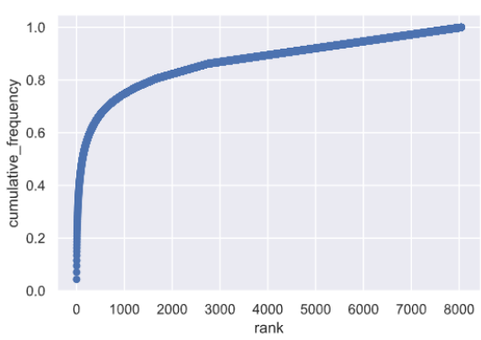
Irony dataset	1 - Ironic	0 - Not ironic
Train	1445	1417
Test	311	473
Validation	465	499

Table 2: Tweets per label on the irony dataset

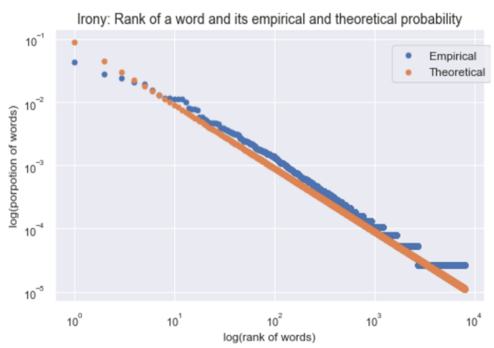
Atheism dataset	0 - none	1 - against	2 - favour
Train	105	273	83
Test	28	160	32
Validation	12	31	9

Table 3: Tweets per label on the atheism dataset

Zipf's law is a discrete probability distribution which gives the probability of encountering a word in a given corpus. This is illustrated in a log-log plot of the frequency of each token against the rank of the given token. In figure 1.b. the empirical and theoretical distributions in the irony dataset is compared. The empirical line is very close to the theoretical straight line, suggesting that the word distribution approximately follows Zipf's Law.



(a) Cumulative plot for the unique tokens in the irony training dataset



(b) Illustration of Zipf's law in a log-log plot for the irony training dataset

Figure 1: Zipf's law

N-grams and maximum likelihood. To understand the relationship between words, one can look at N number of preceding words in a context. This is the basic principle of N-grams, and with this tool one can

calculate the bi-gram probability of the word "x" given the word "y". This process was done for every unique word in the corpus. For the word "user" one can see the probabilities of all words preceding it. Using this simple statistical model alone will generally causes a lot of problems for smaller corpora. In the extreme case it overestimates the probability of words it has only seen once. For example the probability of "user" given "feat" is 100% and "user" given "user" only has a probability of 35%. In the other extreme, the model will underestimate words not seen in the training data, giving these a probability of 0 to occur.

Kneser-Ney. For next word prediction, using the probability of unigrams frequency will lead to skewed results. Kneser-Ney smoothing corrects this skewness by considering the frequency of a unigram in relation to possible words preceding it. This is especially useful when predicting word combinations or sentences not seen before. The project took use of NLTK's 'KneserNeyProbDist' function to estimate the probability of trigrams in the irony dataset.

Word embedding. Another way of associating words with other words of similar meaning is by word embedding. Similar words should appear in similar contexts. With this idea words can then be represented as a vector. This vector has N dimensions, each trying to capture the meaning of the word. The smaller the distance between each vector the more similar the words should be. The word embedding was done for all the biggest available datasets, which resulted in a total of 75000 tweets. With this large amount of data, it was possible to create a meaningful word embedding model, that could predict for example the word "good" being similar to words like "nice" and "great".

3 Annotation

The five group members manually and independently annotated 120 randomly selected tweets provided from the irony dataset, following the set of guidelines provided. The agreement between the annotations was measured by computing different Inter-Annotator Agreement (IAA) coefficients (Table 4 and Table 5), indicating the group's ability to make the same annotation decision for irony.

Obs. agreement	0.6323
S coefficient	0.2656
Scott's π	0.2577
Cohen's κ	0.2923
Fleiss' multi-κ	0.2770

Table 4: Inter-Annotator Agreement coefficient (possibly averaged pair-wise)

The computed IAA coefficients show moderate observed agreement of 0.6323. However, this score can be attributed to chance as the binary labelling makes the probability of annotating the same higher, than if there were more labels. The chance-corrected agreement coefficients takes this into account, and were much lower all under 0.3. This indicates that the observed agreement was most likely by chance, and that irony detection, even for humans, is difficult to detect.

For further measures, pair-wise Cohen's κ was done for each pair of annotator, including the original labels (Table 5).

Group	**ann. 1**	**ann. 2**	**ann. 3**	**ann. 4**	**ann. 5**	**orig.**	**Cohen's κ**
ann. 1	na.	0.5187	0.4053	0.3957	0.0154	0.4034	
ann. 2	0.5187	na.	0.2492	0.3742	0.0188	0.3138	
ann. 3	0.4053	0.2492	na.	0.4046	-0.0532	0.3518	
ann. 4	0.3957	0.3742	0.4046	na.	0.1168	0.7450	
ann. 5	0.0154	0.0188	-0.0532	0.1168	na.	0.1247	
orig.	0.4034	0.3138	0.3518	0.7450	0.1247	na.	
							0.2923

Table 5: Pair-wise Cohen Kappa (κ)-scores for annotator pairs and the average Cohen Kappa score for the group

A few of the pair-wise κ scores indicate moderate agreement, while most indicate fair agreement, according to the Landis and Koch table.[2] Remarkable is the overall low pair-wise κ score between annotator 5 and all others, including negative agreement which indicates more disagreement than agreement between annotators. These IAA coefficients are likely to be the biggest contributor to the low Cohen Kappa score average for the group. The low IAA coefficients can also indicate unclear guidelines that were not uniformly understood by the annotators. Those coefficients are also a reflection of how irony can be hard to detect, even in verbal communication. [1] Detecting it in short texts such as tweets makes the task even more difficult as necessary context could be missing. All these issues make this task not easily reproducible.

It is important to note that the original labels are not necessarily true reflection of the authors intent, as they were also classified by humans following the same guidelines. However, they are still the source of truth and indicator of how well the group members classified the tweets.

The annotation of natural language data is a vital part of the validation when classifying results. The level of difficulty of annotating irony and poor agreement coefficients computed, indicates a bad foundation for achieving accurate classification results.

4 Classification

The classification part of this project is split into two parts: binary and multi-class classification. The multi-class classification is further split into two parts, classification on the stance to the subject atheism and classifying stance with a model trained on all of the different stances in the dataset.

All of the classifications followed the same algorithm:
Firstly tokenize the data, then test against the baseline model, now find the best models in their simplest form (No parameter tuning), followed by hyper parameter optimizing the best simple models and pick the best one on the basis of a set of metrics on the validation data, now pick the best of the tuned models to be the final mode. Finally report metrics for the final model on the test data.

Thus the testing data would only be used for when reporting the final results. Validation data would be used to optimize the models. Training data would only be used to train the models. All of these datasets are disjoint.

4.1 Binary Classification

In the binary classification task, using the described algorithm it was found that all the hyper parameter tuned models were equally bad at classifying irony in a tweet. The Logistic regression was better on the set of metrics that the models were evaluated on. The fact that the models are bad at classifying was not unexpected, since

humans are very bad at identifying irony in textual format as well. Which can also be seen in the poor IAA scores in section 3. It can thus not be expected that a model will outperform humans with world-knowledge, emotional knowledge and contextual knowledge about a given instance where irony occurs. The best model in this task achieved an accuracy of 55% on the test data, which is slightly better than guessing completely at random (uniformly). The f1-score is also around 54% meaning also slightly better than guessing randomly.

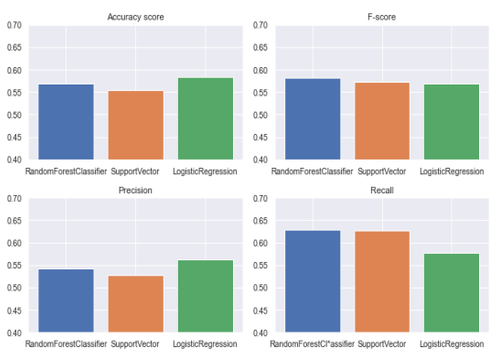
4.2 Multi-class classification on atheism stances

In this task, using the described algorithm, a few complications occurred. Due to the many different classes, the simple models were already confused. It was found that some models just entirely ignored some classes and never classified anything to be that class. This could be caused by imbalanced class population. Leaving out the models that ignored some of the classes in their predictions, left the group with three models with accuracy ranging from 55% to 71%. Then hyper parameter optimization was done to try and optimize the macro averaged recall score, leaving logistic regression as the better model.

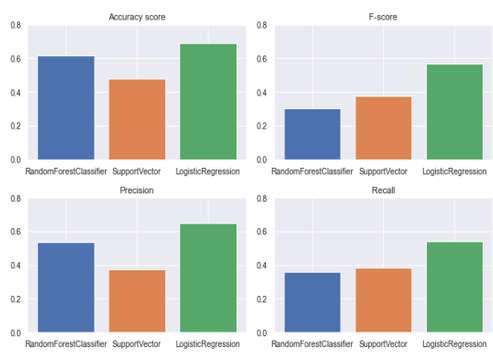
Calculating the metrics for the final model on the test data, it was apparent that the model had an acceptable accuracy on the test data, but not as high precision and recall. This could indicate that the model favors one class over the other and simply predicts the same class all the time. This is possibly due to the imbalance in the training data (Table 6). Looking at the predictions of the final model on the validation data it was apparent that it favored one class over the other.

Class	0	1	2
Predicted	7	40	5
True	12	31	9

Table 6: Confusion table for predicting stance of atheism



(a) Model metrics for irony



(b) Model metrics for atheism

Figure 2: Hyper parameter optimized models on irony and atheism

4.3 Multi-class classification on all the 15 stances in the stance dataset

For this task the mentioned algorithm is used again. Firstly the necessary data had to be produced. All the stance data was merged into one and labels were updated so they ranged from 0 to 14. Unfortunately this created a big imbalance in the data; one class having 13 members, and another class having 354 members. This imbalance clearly impacted the metrics for the simple models.

Only KNearestNeighbor and Decision Tree Classifier managed to include all but one class in their predictions, while most of the other models did not include over half of the classes, raising an issue. Thus the extreme cases were ignored, Support Vector Classifier and Naive Bayes classifier. Due to this imbalance we had to try

and optimize the best simple models to achieve the highest macro averaged recall score. This did not increase the models metrics significantly, though it did increase the inclusion rate in the models' prediction on the validation data. K-nearest neighbor was found to be the best model, not because of its metrics, but because it only excluded class 10 in its validation prediction, which is reasonable since it only has 13 members of its class. An accuracy score of 22% for the KNN and an f1 score of 12.5% were the final metrics: in general really bad metrics, but given the difficulty of the task, the KNN model performs well, and better than if the model were to guess at random. It was also tested what classes, in the validation data, this classifier performs poorly on. It was apparent that this model was very bad at classifying atheism stances compared to models trained and tested only on the atheism dataset.

Subject	Hillary	Abortion	Atheism	Climate	Feminist
Accuracy score	34.8%	13.6%	9.6%	10%	25%

Table 7: Accuracy of KNN model on the separate stance dataset

5 Conclusion and Future Work

Human linguistics, both spoken and written, are highly ambiguous. It is interesting to see if a machine can be trained to understand something as subjective and humane as humor, especially irony, and general intention. Our model for predicting irony performed poorly with an accuracy score of 55%, just a slight bit better than randomly predicting. Taking into account the difficulty the group had of agreeing upon the annotation of irony tweets, it makes sense. For the atheism multiclass classification the imbalance between classes made some predictions which were favoured over others, causing an acceptable accuracy score, but not as high precision or recall.

For the big multiclass classification an even bigger imbalance between classes made it necessary to sacrifice a high accuracy score in trade of not leaving out prediction of some of the classes. Overall it was made clear that the amount of possible classifications and imbalance between these made for a poorly performing model - especially for classes which provided very few input for the training.

For future research a variety of things could be done. First and foremost the tokenizer could be further changed, to accommodate the unique language on social media. The less significant occurrences like 'user', punctuation and stopwords (words such as a, is, an, the, and etc.) could be removed, as these carry minimal to no importance and are available in plenty of open texts, articles, comments etc. These occurrences should be removed so machine learning algorithms can better focus on words which define the meaning/idea of the text - and by that also improve the accuracy of predictions. [3] For the machine learning part another mapping could be done, compressing all the data into one data file with labels 0, 1, 2, where the stance is independent of the subject. It would be interesting to see how a model would perform under these conditions. One could also train a model for each subject and then by using probabilistic modelling find which is most likely to be in what class. This would be easy to implement, though it would have the risk of one class having a consistently higher probability and then it would simply classify any tweet as that class.

Lastly the importance of quality annotation could be ensured by using tweets, where the authors themselves pre-hand labelled their tweets according to the intention hereof. This would ensure a certain standard in more subjective language, due to high differences of perception of annotators in matters such as irony and sarcasm.

References

[1] Roberto González-Ibánez, Smaranda Muresan, and Nina Wacholder. "Identifying sarcasm in Twitter: a closer look". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. 2011, pp. 581–586.

[2] JR. Landis and GG. Koch. "The measurement of observer agreement for categorical data". In: *Biometrics*. 1977, pp. 159–74.

[3] Dinesh Yadav. *NLP: Building Text Cleanup and PreProcessing Pipeline*. Apr. 2020. URL: `https://towardsdatascience.com/nlp-building-text-cleanup-and-preprocessing-pipeline-eba4095245a0`.