NLP Project: Authorship Detection of Twitter Tweets

Clemens Biehl, Daniel Wehner, Fabian Otto, Philipp Kapelle

***Abstract*—In our project we worked on the authorship detection of Twitter tweets. This report gives a short summary of the results which we could produce during the project.**

I. Introduction and Research Problem

Millions of texts and posts are published on social media platforms such as Twitter or Facebook every day. The identification of the authorship is often a crucial task in Natural Language Processing. This might be helpful when checking the authenticity of a post. In this project we therefore aim to classify Twitter Tweets to identify the authors of the tweets and try to answer the following questions:

- Which types of writing-style features are effective for identifying the authorship of online messages?
- Which classifiers perform best for identifying the authorship of online messages?
- To what extent can authorship-identification techniques be applied to online messages across different genres such as politics or sports?

To answer these questions we tried various combinations of features, classifiers and genres such as politics and sports and evaluated these combinations.

II. Data Provisioning

As already mentioned, Twitter will be the source for the data to be analyzed. We focused on tweets of 20 famous politicians, sportsmen, actors and broadcast stars of the English-speaking world since we confined ourselves to english tweets. For instance, we collected tweets from the following accounts of politicians:

realDonaldTrump, BarackObama, ChuckGrassley, RepJaredPolis, BorisJohnson, clairecmc, ChrisChristie, jahimes, jeremycorbyn, CarolineLucas, David_Cameron, BernieSanders, RonPaul, SpeakerRyan, mike_pence, DavidLammy, timfarron, Ed_Miliband, ChukaUmunna, tom_watson

The number of politicians' tweets collected per account is 1,000, which makes 20,000 tweets in total. When guessing the authors randomly, we would get an accuracy of approximaltey 5 % (since the training data consists of tweets from 20 different authors). This is our baseline. We should have an accuracy better than 5 %.

III. Pipeline

The general process for the authorship identification follows a usual NLP pipeline approach as depicted in Figure 2. The single steps will be described further in the following sections. The *Data Provisiong* was described in the last section. The preprocessing depends on the features which are used, e. g. the contextuality measure requires part-of-speech tagging.

Data Provisioning: For information see section II

Feature Engineering: Please see section IV

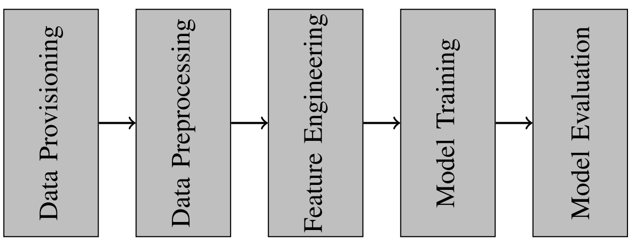
Model Training and Evaluation: Please refer to section V

TU Darmstadt, WS 2017/2018.

Fig. 1. JSON representation of a tweet written by Barack Obama that was crawled from Twitter


```
{
   "in_reply_to_status_id_str":null,
   "in_reply_to_status_id":null,
   "coordinates":null,
   "created_at":"Mon Jan 15 14:46:02 +0000 2018",
   "truncated":false,
   "in_reply_to_user_id_str":null,
   "source":"<a href=\"http://twitter.com/download/
     iphone\" rel=\"nofollow\">Twitter for
     iPhone<\/a>",
   "retweet_count":367963,
   "retweeted":false,
   "geo":null,
   "in_reply_to_screen_name":null,
   "is_quote_status":false,
   "entities":{
      "urls":[

      ],
      "hashtags":[

      ],
      "user_mentions":[

      ],
      "symbols":[

      ]
   },
   "full_text":"Dr. King was 26 when the Montgomery
      bus boycott began. He started small, rallying
      others who believed their efforts mattered,
      pressing on through challenges and
      doubts to change our world (...)",
   "id_str":"952914779458424832",
   "in_reply_to_user_id":null,
   "display_text_range":[
      0,
      279
   ],
   "favorite_count":1393878,
   "id":952914779458424832,
   "place":null,
   "contributors":null,
   "lang":"en",
   "user":{
      "id_str":"813286",
      "id":813286
   },
   "favorited":false
}
```


Fig. 2. Visualization of the pipeline built for the authorship identification. After providing the data by the crawler, the data is preprocessed. The feature extraction phase takes care of creating useful features for classification for the training of the model. The last step is the evaluation of the model.



IV. Feature Engineering

The feature engineering phase is crucial for the success of the project since the performance of the classifiers depends significantly on the quality of the features used. Table I summarizes the features which we have taken into account when training the classifiers.

TABLE I
List of features which were considered for training the classifiers.

Feature	Explanation
Total number of chars	How long is the tweet
Emoticon ratio	Proportion of emoticons in the text.
Number of hashtags	How many hashtags are used in the tweet.
Word frequencies	Which words are used frequently
Lexical diversity	How rich is the vocabulary of the author?
Contextuality Measure	Score between 0 and 100 (0 very context dependent = many pronouns, adverbs, ...; 100 not content dependent = many nouns, ...)
Exclamation Ratio	How many exclamation marks are used?
Superlative Ratio	Proportion of superlatives in the tweet?
PastVsFuture	Ratio of verbs in past tense/present tense

With emoticons playing a central role in social media they represent a good feature to consider. Therefore, the emoticon ratio (ER) is calculated (the number of tokens which represent emoticons divided by the total number of tokens in the text).

Other features like sentence features did not perform well since twitter texts are very short.

$$\text{ER} = \frac{\text{\# of emoticon tokens}}{\text{\# of tokens}} \quad (1)$$

Also very characteristic for specific authors is the number of hashtags they use when writing a tweet, the word frequencies (does the author prefer specific words over other words) and the lexical diversity (also known as type-token ratio [TTR] which analyzes how many different words are used in the tweet)

$$\text{TTR} = \frac{V(N)}{N} \quad (2)$$

Other measures for lexical diversity/vocabulary richness:

$$\text{Yule's } K = C\left[-\frac{1}{N} + \sum_{m=1}^{m_{max}} V(m,N)\left(\frac{m}{N}\right)^2\right] \tag{3}$$

$$\text{Simpson's } D = \sum_{m=1}^{m_{max}} V(m,N)\frac{m}{N}\frac{m-1}{N-1} \tag{4}$$

$$\text{Herdan } V_m = \sqrt{\sum_{m=1}^{m_{max}} V(m,N)\left(\frac{m}{N}\right)^2 - \frac{1}{V(N)}} \tag{5}$$

$$\text{Sichel's } S = \frac{V(1,N)}{V(N)} \tag{6}$$

$$\text{Honore's } R = 100\frac{\log(N)}{1-\frac{V(1,N)}{V(N)}} \tag{7}$$

$$\text{Brunet's } W = N^{V-c} \quad \text{usually } c = 0.17 \tag{8}$$

$$\text{Uber Index} = \frac{\log(N)^2}{\log(N) - \log(V(N))} \tag{9}$$

N	Length of text
$V(N)$	Size of vocabulary
$V(m,N)$	Number of words in N occuring m times
$V(1,N)$	Number of **Hapax Legomena**
$V(2,N)$	Number of **Hapax Dislegomena**
m_{max}	maximal frequency

Another feature of interest is the contextuality measure which indicates how context-dependent a text is. The contextuality measure produces values between 0 and 100. A value of 0 indicates a very context-dependent text which contains many pronouns, adverbs, etc. The higher the value the less context-dependent the text is (the text is then said to be *formal* as opposed to *contextual*). This is the formula which computes the score:

$$F = \frac{n + a + p + d - pr - v - ad - if + 100}{2} \tag{10}$$

n	noun frequency
a	adjective frequency
p	preposition frequency
d	determiner frequency
pr	pronoun frequency
v	verb frequency
ad	adverb frequency
if	interjection frequency

V. Evaluation

As a basis we used the `WekaTwitterSentimentDemo` which we found in the DKPro TC GitHub Repository resulting in an accuracy of approximately $13\,\%_{\text{Naive Bayes}}$, approximately $8.5\,\%_{\text{Random Forest}}$ and approximately $8\,\%_{\text{Logistic}}$ (Used features in the demo: EmoticonRate and Number of Hashtags, Number of Tokens per Sentence). We used this as the baseline and added further features. The evaluation is based on a test set of 10% of the tweets per genre. The results for different features and classifiers are summarized in tables II (**Naive Bayes**), III (**Random Forest**) and IV (**Logistic Regression**) and in figures 3 (**Naive Bayes**), 4 (**Random Forest**) and 5 (**Logistic Regression**) respectively.

Fig. 3. Evaluation results for different feature setups. Results for **Naive Bayes** classifier

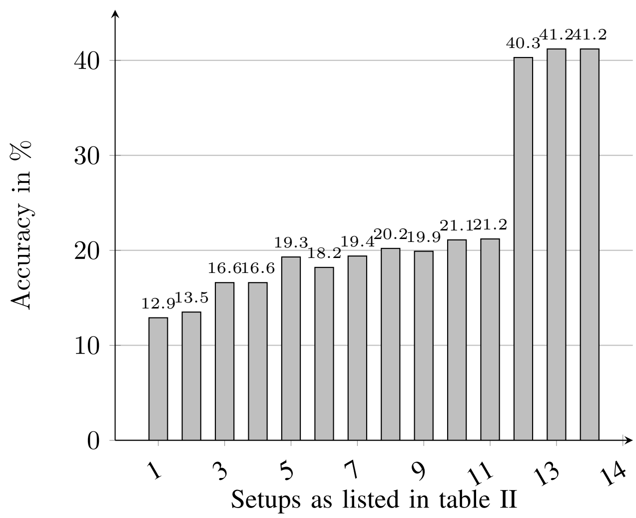


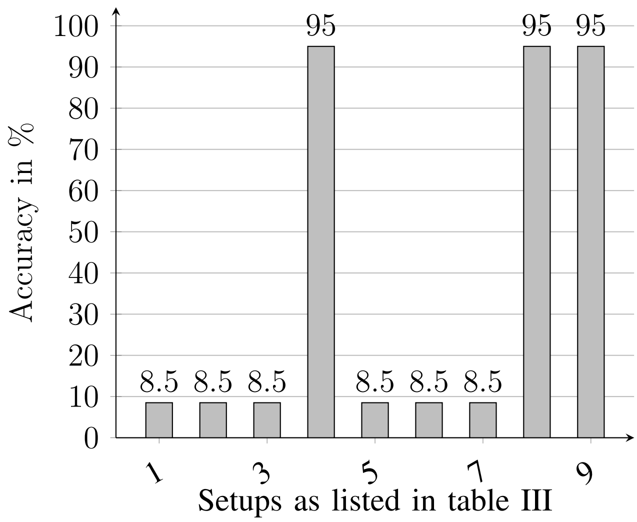
TABLE II
EVALUATION RESULTS FOR DIFFERENT FEATURE SETUPS ORDERED BY INCREASING ACCURACY. CLASSIFIER = **NAIVE BAYES**

Naive Bayes		
Nr.	**Setup**	**Accuracy**
1	`WekaTwitterSentimentDemo`	**12.88 %**
2	TTR (Type-Token-Ratio)	**13.48 %**
3	TTR + Contextuality Measure	**16.60 %**
4	TTR + UpperCase + Alphabetic + Digits + WhiteSpaces + TabSpaces	**16.59 %**
5	TTR + Contextuality Measure + Exclamation Ratio	**19.25 %**
6	TTR + Contextuality Measure + Exclamation Ratio + Superlative Ratio	**18.19 %**
7	TTR + Contextuality Measure + Exclamation Ratio + PastVsFuture	**19.38 %**
8	TTR + Contextuality Measure + Exclamation Ratio + PastVsFuture + Avg Nr of Chars per Sentence	**20.22 %**
9	TTR + Contextuality Measure + Exclamation Ratio + PastVsFuture + Avg Nr of Chars per Sentence + Avg Nr of Chars per Token + Nr of Chars + Nr of Sentences + Nr of Tokens	**19.90 %**
10	TTR + Contextuality Measure + Exclamation Ratio + PastVsFuture + Avg Nr of Chars per Sentence + UpperCase + Alphabetic + Digits + WhiteSpaces + TabSpaces	**21.06 %**
11	TTR + Contextuality Measure + Exclamation Ratio + PastVsFuture + Avg Nr of Chars per Sentence + UpperCase + Alphabetic + Digits + WhiteSpaces + TabSpaces + Yule's K + Herdan V_m	**21.19 %**
12	TTR + Contextuality Measure + Exclamation Ratio + PastVsFuture + Avg Nr of Chars per Sentence + UpperCase + Alphabetic + Digits + WhiteSpaces + TabSpaces + N-Grams (up to trigrams)	**40.26 %**
13	TTR + Contextuality Measure + Exclamation Ratio + PastVsFuture + Avg Nr of Chars per Sentence + UpperCase + Alphabetic + Digits + WhiteSpaces + TabSpaces + N-Grams (up to trigrams) + POS N-Grams (up to trigrams)	**41.20 %**
14	TTR + Contextuality Measure + Exclamation Ratio + PastVsFuture + Avg Nr of Chars per Sentence + UpperCase + Alphabetic + Digits + WhiteSpaces + TabSpaces + N-Grams (up to trigrams) + POS N-Grams (up to trigrams) + SpellingErrorRatio	**41.20 %**

TABLE III
EVALUATION RESULTS FOR DIFFERENT FEATURE SETUPS ORDERED BY INCREASING ACCURACY. CLASSIFIER = **RANDOM FORESTS**

Random Forest		
Nr.	**Setup**	**Accuracy**
1	`WekaTwitterSentimentDemo`	**8.49 %**
2	TTR (Type-Token-Ratio)	**8.49 %**
3	TTR + Contextuality Measure	**8.49 %**
4	TTR + UpperCase + Alphabetic + Digits + WhiteSpaces + TabSpaces	**95.02 %**
5	TTR + Contextuality Measure + Exclamation Ratio	**8.49 %**
6	TTR + Contextuality Measure + Exclamation Ratio + Superlative Ratio + PastVsFuture	**8.49 %**
7	TTR + Contextuality Measure + Exclamation Ratio + Superlative Ratio	**8.49 %**
8	TTR + Contextuality Measure + Exclamation Ratio + Superlative Ratio + Nr of Tokens + Character Features + N-Grams	**95.08 %**
9	TTR + Contextuality Measure + Exclamation Ratio + Superlative Ratio + Nr of Tokens + Character Features + N-Grams + POS-NGrams	**95.03 %**

Fig. 4. Evaluation results for different feature setups. Results for **Random Forest** classifier



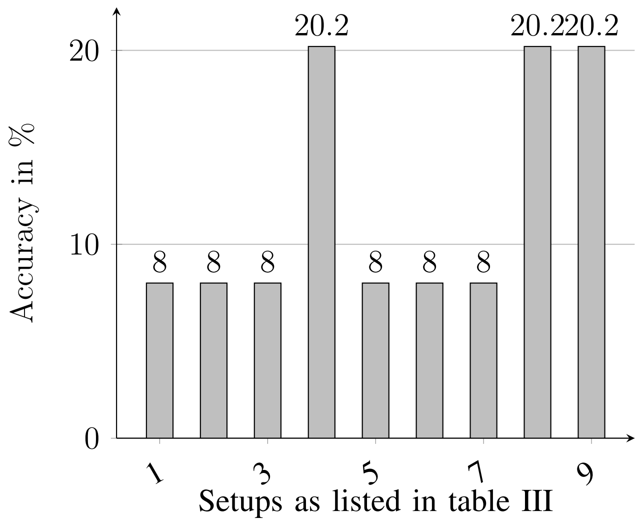
VI. DEEP LEARNING

Extracting the features is very tedious and it is more or less a trial and error process. There are many

TABLE IV
EVALUATION RESULTS FOR DIFFERENT FEATURE SETUPS ORDERED BY INCREASING ACCURACY. CLASSIFIER = **LOGISTIC REGRESSION**

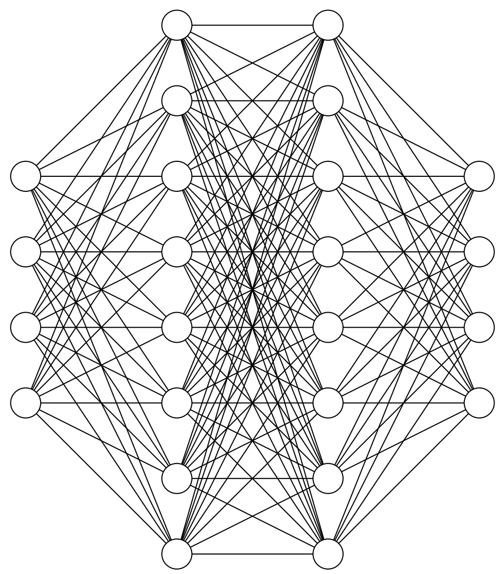
Logistic Regression		
Nr.	**Setup**	**Accuracy**
1	`WekaTwitterSentimentDemo`	**7.96 %**
2	TTR (Type-Token-Ratio)	**7.96 %**
3	TTR + Contextuality Measure	**7.96 %**
4	TTR + UpperCase + Alphabetic + Digits + WhiteSpaces + TabSpaces	**20.20 %**
5	TTR + Contextuality Measure + Exclamation Ratio	**7.96 %**
6	TTR + Contextuality Measure + Exclamation Ratio + Superlative Ratio + PastVsFuture	**7.96 %**
7	TTR + Contextuality Measure + Exclamation Ratio + Superlative Ratio	**7.96 %**
8	TTR + Contextuality Measure + Exclamation Ratio + Superlative Ratio + Nr of Tokens + Character Features + N-Grams	**20.20 %**
9	TTR + Contextuality Measure + Exclamation Ratio + Superlative Ratio + Nr of Tokens + Character Features + N-Grams + POS-NGrams	**20.20 %**

Fig. 5. Evaluation results for different feature setups. Results for **Logistic Regression** classifier



combinations of features that have to be taken into account and which have to be evaluated. A remedy for that is for example a **Deep Learning** approach. Such approaches have become very famous recently and also for Natural Language Processing there are several application areas for such methods. Deep Learning is capable of automating this combersome process by making use of several layers that are responsible for feature extraction and transformation. The results of one layer represent the input of the next layers. Very often **Artificial Neural Networks (ANN)** are used in such cases.

Fig. 6. A simple neural network with two hidden layers.



VII. COMPONENT DIAGRAM

VIII. CONCLUSION

The experiments showed that character-based linguistic style features such as the frequencies of upper case, alphabetic, digit and white- and tabspace characters are good indicators for an author's style in writing tweets. Likewise the Contextuality Measure and PastVsFuture features increased accuracy. In contrast, sentence-based features such as the number of sentences or the average number of characters in a sentence decreased accuracy, which can be explained by the general brevity of tweets and the fact that tweets often do not include clear sentence boundaries. Some of the measures for lexical diversity/vocabulary richness did not contribute to the accuracy, therefore only Yule's K and Herdan V_m were used. Using N-Grams and Part-of-Speech N-Grams significantly improved the accuracy, presumably because they incorporate the

author's specific vocabulary.
The final accuracy of 95.08% using a random forest classifier and the features shown in table III (column 8) is remarkably high for the task of authorship identification and might not generalize well to other domains, since the N-Grams are very specific with respect to the training data.
Future experiments will have to investigate if the features used here can be applied in and across different domains. Furthermore, deep learning has to be considered for including an implicit feature extraction in the classifier training.

REFERENCES

[1] Zheng, Rong and Li, Jiexun and Chen, Hsinchun and Huang, Zan. A Framework for Authorship Identification of Online Messages: Writing-Style Features and Classification Techniques. *Journal of the American Society for Information Science and Technology*, vol. 57, no. 3, pp. 378-393, 2006.
[2] Hout, Roeland and Vermeer, Anne. Comparing measures of lexical richness. *Modelling and assessing vocabulary knowledge*, pp. 93-116, 2007.
[3] Fissette, Marcia. Author identification in short texts. 2010.
[4] Green, Rachel M. and Sheppard, John W. Comparing Frequency- and Style-Based Features for Twitter Author Identification. *AAAI Press*, 2013.
[5] Heylighen, Francis; Dewaele, Jean-Marc. Variation in the Contextuality of Language: An Empirical Measure. *Foundations of Science*, vol. 7, no. 3, pp. 293-340, September 2002.
[6] Tanaka-Ishii, Kumiko and Aihara, Shunsuke. Computational Constancy Measures of TextsYule's K and Rnyi's Entropy. *Computational Linguistics* vol. 41, no. 3, pp. 481-502, September 2015.
[7] Tweedie, Fiona J. and BaayenHow, R. Harald. Variable May a Constant be? Measures of Lexical Richness in Perspective. *Computers and the Humanities* vol. 32, no. 5, pp. 323-352, September 1998.
[8] Bhatia Archna et al. TweetNLP, Carnegie Mellon. http://www.cs.cmu.edu/~ark/TweetNLP