University of Twente

Master Thesis

Trust in automated decision making

How user's trust and perceived understanding is influenced by the quality of automatically generated explanations

Author:
Andrea Papenmeier

Supervisor:
Dr. Christin Seifert
Dr. Gwenn Englebienne

December 30, 2018

UNIVERSITY
OF TWENTE.

Abstract

1 Introduction

asd [1] asd [2]

2 Background

Intro

2.1 Opacity in AI

Intro

2.1.1 Opacity Sources

asd

2.1.2 Application Areas

asd

2.1.3 Examplary Failures

asd

2.1.4 Regulations

asd

2.1.5 Use Case Scenario

asd definition of offensive language

2.2 Explanations

Intro

2.2.1 Explanation Types

asd

2.2.2 Social Sciences

asd

2.2.3 Psychology

asd

2.3 Explanations in AI

Intro

2.3.1 Explanation Focus

asd

2.3.2 Explanations for Texts

asd

2.4 Trust in AI

Intro

2.4.1 Persuasiveness

asd

2.4.2 Accountability

asd

2.4.3 Trust

asd

2.5 Summary

Summary
Hypotheses

3 Dataset

Intro

3.1 Dataset Selection

Few datasets with offensive language texts are publicly available. Table 1 presents an overview of four available datasets, their sizes and class balances. While the dataset of SwissText has the most fine-grained labelling of its data

Corpus	**Size**	**Classes**	
Davidson[1]	25,000	hate speech	6%
		offensive	77%
		neither	17%
Imperium[2]	3,947	neutral	73%
		insulting	27%
Analytics Vidhya[3]	31,962	hate speech	7%
		no hate speech	93%
SwissText[4]	159,570	toxic	10%
		severe_toxic	1%
		obscene	5%
		threat	0.3%
		insult	5%
		hate speech	1%
		neither	72.7%

Table 1: Publicly available datasets for offensive language texts

points, details on how the labels were assigned (i.e. number of annotators, inter-annotator agreement score, definition of the classes) are not available. The same holds for the datasets of Analytics Vidhya and Imperium.
In contrast, Davidson's datasets comes with a description of how the data points were collected, how the classes are defined, and uses at least three annotators per text. Furthermore, Davidson's dataset contains the most data points labelled as offensive: roughly 20750 Tweets fall into this category, while the Analytics Vidhya dataset contains 2240 hate speech texts, SwissText 1600, and Imperium 1000.
Throughout the literature, different definitions of hate speech and offensive language are given. For using a dataset in a user study with the scenario of a social media administrator, the definition of the label has to be clear. We therefore chose to work with the dataset of Davidson et al., as it offers the most detailed description of its labels and how the labels were obtained.

3.2 Dataset Construction

The original dataset was collected by Davidson et al. [3] for their research on defining and differentiating hate speech from offensive language. They con-

structed a dataset with offensive Tweets and hate speech by conducting a keyword search on Twitter, using keywords registered in the hatebase dictionary[5]. The timelines of Twitter users identified with the keyword search were scraped, resulting in a dataset of over 8 million Tweets. They selected 25 000 Tweets at random and had at least 3 annotators from Figure Eight[6] (formerly Crowd Flower) who labelled each Tweet as containing hate speech, offensive language, or neither. They reached an inter-annotator agreement of 0.92 [3]. The dataset is publicly available on GitHub[7].
The biggest class in the dataset are the offensive language Tweets (77%), while non-offensive Tweets represent 17%, and hate speech 6% of the dataset.
For our research, we are only interested in offensive and not offensive Tweets. We therefore excluded Tweets labelled as hate speech for the further construction of our dataset. We produced a balanced dataset by selecting only Tweets with the maximum inter-annotator agreement from each of the two remaining classes, and randomly drew Tweets from the bigger class (offensive Tweets) until the size of the subset was equal to the size of the smaller class (non-offensive Tweets). Table 2 presents statistical information about the resulting dataset.

	Not Offensive Class	**Offensive Class**
Size (absolute)	4,162	4,162
Size (relative)	50.00%	50.00%
Total words	58,288	61,504
Unique words	6,437	9,855
Average words per Tweet	14.00	14.78

Table 2: Statistical characteristics of the constructed dataset

3.3 Dataset Preprocessing

Tweets exhibit some special characteristics. First, the maximum length of a single Tweet is 140 characters. Twitter doubled the length in November 2017, yet the dataset was collected before this data and therefore contains only Tweets of 140 characters or shorter. Twitter users found creative ways to make use of the 140 characters given, leading to the usage of short URLs instead of original URLs [13], intentional reductions of words (e.g. “nite” instead of “night”) [13], abbreviations [5], emojis [4] [12] and smilies [11] [7].
Furthermore, social media content can be unstructured, with word creations that are non in standard dictionaries, like slang words [5] [12], intentional repetitions [13] [6] [9] [10] (e.g. “hhheeeey”), contractions of words [11] [6], and spelling mistakes. Although those new word formations do not appear in the dictionary, they are “intuitive and popular in social media” [8].

[5] https://www.hatebase.org
[6] https://www.figure-eight.com
[7] https://github.com/t-davidson/hate-speech-and-offensive-language

On Twitter, it is custom to mention other users within a Tweet by adding "@"+username [13] [9] [12] [10], retweeting (i.e. answering to) a Tweet [13] [6], and summarizing a Tweet's topic with "#"+topic [13] [12].
Other problems in text mining are the handling of stop words [13] [4] [5], language detection [13], punctuation [4] [6] [9], negation [12], and case folding [4] [5] [10].
Researchers have developed different strategies for preprocessing Tweets. One possible approach is to simply remove URLs, username, hashtags, emoticons, stop words, or punctuation [13] [4] [6] [9] [5] [12]. A reason to eliminate those tokens can be that they assumably do not hold information relevant to the classification goal [6]. Words that only exist for syntactic reasons (this concerns primarily stop words) can be omitted when focussing on sentiment or other semantic characteristics [4]. Mentions of other users are likewise not informative for sentiment analysis and are often removed from the texts [13] [12]. Depending on the dataset size, normalising the texts strongly by removing punctuation and emojis, as well as lowercasing the texts, can decrease the vocabulary size [4]. Especially on Twitter with its restricted text size, users tend to use shortened URLs. Short URLs have a concise, but often cryptic form, and redirect to the website with the original, long URL. While website links can encode some information on a topic, this information is lost when using a shortened URL. Removing the shortened URLs without replacement can be a step in preprocessing Tweets [13].
Rather than removing tokens, they can also be replaced by a signifier token, e.g. a complete link by "<<<hyperlink>>>" [7]. In Tweets, such signifier tokens are used for mentions of usernames [11] [7] [10], URLs [11] [7] [10], smilies [7] or negations [11]. Using signifier tokens eliminates some information, i.e. which user was mentioned or which website was linked, but retains the information that a mention or link exists. Tokens can also be grouped by using signifier tokens, i.e. tokens with similar content are summarised with a single token. [7] uses this technique to group smilies with similar sentiment and Twitter usernames related to the same company.
Case folding is often addressed by converting Tweets to lower case [4] [7] [5].

The following preprocessing steps are taken in chronological order:

1. Conversion of all texts to lower cases
2. Replacement of URLs by a dummy URL ("URL")
3. Replacement of referenced user names and handles by a dummy handle ("USERNAME")
4. This dataset encodes emojis in unicode decimal codes, e.g. "😀" for a grinning face. In order to keep the information contained in emojis, each emoji is replaced by its textual description (upper cased and without whitespaces to ensure unity for tokenizing)[8].

[8] https://www.quackit.com/character_sets/emoji/

5. Resolving contractions such as “we’re” or “don’t” by replacing contractions with their long version[9].
6. This dataset uses a few signifiers such as “english translation” to mark a Tweet that has been translated to English, or “rt” to mark a Retweet (i.e. a response to a previous Tweet). Since those information have been added retrospectively, we discard them here and delete the signifiers from the texts.
7. Replacement of all characters that are non-alphabetic and not a hashtag by a whitespace
8. Replacement of more than one subsequent whitespace by a single whitespace
9. Tokenization on whitespaces

After training the classifiers, the URL and username tokens are replaced by a more readable version (“http://website.com/website” and “@username”, respectively) to make it easier for participants of the user study to envision themselves in the scenario of a social media administrator reading real-world Tweets. Replacing the tokens by their original URLs and usernames would give the participants more information than the classifiers had; we therefore chose to use a dummy URL and username.

Following the preprocessing steps, the following Tweet is processed from its original form:

```
"@WBUR: A smuggler explains how he helped fighters along the

"Jihadi Highway": http://t.co/UX4anxeAwd"
```

into a cleaned version:

```
@username a smuggler explains how he helped fighters along the

jihadi highway http://website.com/website
```

[9] https://en.wikipedia.org/wiki/Wikipedia:List_of_English_contractions

4 Design

Intro

4.1 Classifier

Intro

Good System L2X

Medium System Logistic Regression with binary (1 / -1) coefficients

Bad System Inverse L2X

4.2 Explanations

Intro

Good System L2X mutual information

Medium System randomly choosing k words from the words with positive (offensive) or negative (not offensive) class

Bad System Inverse good system

4.3 Graphical User Interface

asdasdasd

4.4 Subset Sampling

For evaluating the different system-explanation conditions, users have to experience the system. However, it is not feasible to present them with the complete testeset, since it has a size of 1665 Tweets. Consequently, a subset of Tweets needs to be drawn from the testset, with a size that a human observer can understand and process within the time frame of a user study.
We furthermore aim to find 10 suitable subsets and assign participants randomly to one of the subsets, in order to reduce possible side effects from biases specific to single Tweets.
There are several requirements for the subsamples, originating from the conflict of reducing the sample for a human observer, yet still yielding a good representation of the testset and classifier:

- A class balance of the true labels similar to the testset,

- a balance of correctly to incorrectly classified data points similar to the classifier's performance on the complete testset,
- no overlap of Tweets within the set of 10 subsets,
- a feature distribution as close to the feature distribution in the complete testset.

We set the subsample size to 15 Tweets, which is enough to show accuracies to the first decimal place, yet assumably not too much to process for an observer in a user study.
To create a subset, 15 data points are randomly drawn from the testset.
First, the class balance of the subset is calculated. The difference to the class balance of the whole testset needs to be smaller than 0.1.
Additionally, for each classifier in the user study, the prediction accuracy on the subset is compared to the prediction accuracy on the complete testset. If, for all classifiers, the difference is smaller than 0.1, the next check is performed.
To ensure the uniqueness of the subsets, the randomly drawn Tweets are compared with the content of previously found subsets. The subset is only accepted if none of the contained Tweets appear in any previously found subset.
In the last step, the feature distribution of the subset is tested against the features of the complete testset using the *Kullback-Leibler Divergence* (KLD) metric. As the focus is directed towards the explanations (i.e. the highlighted words within a Tweet), only the explanations are used to examine the feature distribution. First, the feature distribution of the complete testset is calculated by constructing a word vector with tuples of words and their respective word counts. The word counts are divided by the total amount of words in the set, such that the sum of regularised counts equals 1. Next, a copy of the word vector is used to count and regularise the word frequencies in the subset. The result are two comparable vectors, yet the vector of the subset is very likely to contain zero counts for words that appear in the complete set but were never selected as explanation in the subset. Since the KLD uses the logarithm, it is undefined for zero counts. We use Laplace smoothing with k=1 to handle zero counts. For each classifier, the KLD is calculated and summed to a total divergence score for the subset.
We generate a quantity of 100 such subsets and order them by their KLD sum. The 10 subsets with the smallest score are chosen as the final set of subsets.

5 Experiment 1: Explanation Evaluation

5.1 Method

Intro

5.2 Results

asdasdasd

6 Experiment 2: Trust Evaluation

Intro

6.1 Method

Intro

Participants

- amount, mean age, SD age
- recruitment method
- exclusion criteria
- compensation for participation

Apparatus A paragraph about the experiment setup (physically), system requirements and technology used. For example the pixel dimensions of screenshots.

Procedure

- tasks / survey items
- ordering of tasks

Design & Analysis One paragraph for experiment design (statistically).

One paragraph for statistical analysis.

- data points per participant and in total
- statistical test
- corrections / disqualifications

6.2 Results

Intro

asdasdasd

7 Discussion

8 Conclusion

asd

References

[1] Hiva Allahyari and Niklas Lavesson. User-oriented assessment of classification model understandability. In *11th scandinavian conference on Artificial intelligence*. IOS Press, 2011.

[2] Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. " what is relevant in a text document?": An interpretable machine learning approach. *PloS one*, 12(8):e0181142, 2017.

[3] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, ICWSM '17, pages 512–515, 2017.

[4] T. Ghorai. An information retrieval system for fire 2016 microblog track. In *Workshop Proceedings working notes of Forum for Information Retrieval Evaluation (FIRE)*, volume 1737 of *CEUR '16*, pages 81–83. CEUR-WS.org, 2016.

[5] Priya Gupta, Aditi Kamra, Richa Thakral, Mayank Aggarwal, Sohail Bhatti, and Vishal Jain. A proposed framework to analyze abusive tweets on the social networks. *International Journal of Modern Education and Computer Science*, 10(1):46, 2018.

[6] I Hemalatha, GP Saradhi Varma, and A Govardhan. Preprocessing the informal text for efficient sentiment analysis. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, 1(2):58–61, 2012.

[7] Leonard Hövelmann, Stockholmer Allee, and Christoph M Friedrich. Fasttext and gradient boosted trees at germeval-2017 on relevance classification and document-level polarity. *Shared Task on Aspect-based Sentiment in Social Media Customer Feedback*, page 30, 2017.

[8] Xia Hu and Huan Liu. Text analytics in social media. In *Mining text data*, pages 385–414. Springer, 2012.

[9] Joaqun Padilla Montani. Tuwienkbs at germeval 2018: German abusive tweet detection. *Austrian Academy of Sciences, Vienna September 21, 2018*, 2018.

[10] Kristian Rother, Marker Allee, and Achim Rettberg. Ulmfit at germeval-2018: A deep neural language model for the classification of hate speech in german tweets. *Austrian Academy of Sciences, Vienna September 21, 2018*, 2018.

[11] Jasmina Smailović, Miha Grčar, Nada Lavrač, and Martin Žnidaršič. Predictive sentiment analysis of tweets: A stock market application. In *Human-computer interaction and knowledge discovery in complex, unstructured, Big Data*, pages 77–88. Springer, 2013.

[12] Hajime Watanabe, Mondher Bouazizi, and Tomoaki Ohtsuki. Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection. *IEEE Access*, 6:13825–13835, 2018.

[13] Guang Xiang, Bin Fan, Ling Wang, Jason Hong, and Carolyn Rose. Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1980–1984. ACM, 2012.