#MeTooMaastricht

Glazunov, Misha Devrim, Emre Jaramillo Lopez, William Bauer, Tobias
Mohan, Balaganesh

June 26, 2019

1 Introduction

As one of the most influential social movements in recent years, #MeToo has enabled sexual harassment to rise to the surface that usually does not get the attention required [1]. There are various types of sexual harassment such as verbal, physical or non-verbal issues in real life and unfortunately, those are some of the most under-reported criminal offenses. Most victims may not be willing to go to the police or reveal these issues on social media or even people around, although they are affected mentally or physically or both. The reasons for this can be manifold. One of the reasons can be, for example, the feeling of shame or embarrassment [2].

In this nonprofit text mining project, #MetooMaastricht, we aim to help sexual harassment victims in the city of Maastricht, the Netherlands. Therefore, we come up with the idea of an intelligent tool, namely a chatbot, which can retrieve crucial information from victims' texts such as the types of harassment as well as the time and location of the event in order to suggest the best set of actions.

Bearing in mind the previous studies against sexual harassment and text mining techniques; our main research questions were as follows:

- How can we best design and implement an intelligent chatbot in order to advise people affected by harassment cases?
- How can we successfully classify different types of harassment cases based on short texts by using text classification techniques?
- Can we extract time and location information from these texts?
- How can we use the information extracted from our models in our final product, a chatbot, for proper guidance to victims?

2 Previous work

Most of this work is based on the concepts and techniques used in the domain of natural language processing (NLP), in this section, we set a theoretical framework of our project, namely language modelling approaches, language representations, the used NLP techniques and knowledge bases.

2.1 Generative language models

2.1.1 N-grams

One of the simplest language models is based on n-grams [3]. An n-gram is a sequence of tokens which may represent characters, morphemes, words, etc. Each of such sequences is assigned with the probability of its occurrence. It provides a numeric value that may be used to evaluate the context or to generate the next most probable token. This approach was widely used in NLP especially for speech recognition, spell-checking and set phrase identification.

2.1.2 Recurrent Neural Networks

First successful attempts of applying neural networks to language modelling were based on recurrent neural networks (RNNs) [4] and subsequently mainly on long short-term memory units (LSTMs)[5]. Architecturally, they are divided into two parts: encoder and decoder. The Encoder allows capturing the sequential nature of input sequences by memorizing the context and extracting the most important features of the whole sequence. The Decoder, on the other hand, attempts to generate the output sequence based on the hidden state computed by the encoder.

2.1.3 Attention Models

One of the main issues with the RNN encoder-decoder approach is that it tries to encode the whole sequence into a feature vector that is afterwards used by the decoder. Attention mechanism [6] solves this by allowing the decoder to get the hidden states of the encoded sequence by taking a weighted average of the input hidden context. This approach helps taking into consideration different parts of the sequence with the appropriate emphasis on these parts that results in a better contextual language modelling.

2.1.4 Self-Attention

Originally, attention models have been applied to machine translation where two distinct sequences are handled and aligned [6]. Subsequently, self-attention models were introduced that relate different parts of a sequence. They showed good results in several domains such as machine reading and text summarization [7], [8].

2.1.5 Transformers

Another main problem with RNNs is the intrinsic sequential nature of the processing which results in slow model training and difficulty to parallelize the optimization procedure. As a result, a new network architecture has recently been purposed, namely, the Transformer [9]. It is based on the self-attention mechanism and allows to discard RNNs completely.

2.2 Discriminative language models

Both n-grams [10] and RNNs [11] can also be used to obtain discriminative language models. A particular example of this kind of models are conditional random fields (CRFs). Despite the generative models they model conditional probability distributions. They are widely applied in the classical methods of named entity recognition [12] and part-of-speech tagging [13] tasks.

2.3 Pretrained universal language models

A recent research described in [14] demonstrates the feasibility of the approach when a language model is first pretrained on a big volume of unlabeled texts in a semi-supervised manner [15] that can be subsequently fine-tuned to a specific NLP task with the state-of-the-art performance results. This method introduces a novel way of dealing with text processing when the so-called universal language model codifies all the morphological, syntactic and semantic features that can be subsequently used in natural language understanding.

2.4 Language representations

Besides modeling the probability distribution of the language tokens, there are several types of language representation. We briefly describe the most influential ones: word embeddings, word-piece embeddings, and document/paragraph embeddings.

2.4.1 Word embeddings

The motivation behind finding better representations for categorical data comes from the limitations of the traditional use of one-hot encoding mapping of categorical variables, where each category is mapped to a high N-dimensional vector consisting of a single "one" representing a specific value in the variable category, and N-1 zeroes alongside representing the other possible values for the same variable. To overcome the limitations present in one hot encoding representations, approaches like Word2vec models have been used in NLP. These models create a dense high dimensional vector representation for each unique word in the corpus of a text input. The vectors obtained are positioned in the vector space such that words that share the same context or are similar are close to one another in that space [16]. The two main model architectures used in the Word2vec algorithm are: continuous bag-of-word (CBOW) and skip-gram (SG) models. The main difference between the two of them is that while CBOW takes multiple context of each word as inputs and tries to predict the word corresponding to its context, skip-gram uses the target word to predict the context [17].

2.4.2 Word-piece embeddings

Another possible kind of representation is a word-piece [18]. It is a sub-word unit which provides a good balance between character and word representations. Wordpiece units are especially important in languages with rich morphological structure and it is linguistically close to the notion of morpheme.

2.4.3 Document/paragraph embeddings

Paragraph or document vector (Doc2vec) is the extended version of Word2vec such that Word2vec learns the d-dimensional representation of words while Doc2vec aims to learn projection of documents into dimensional space. For this purpose, the authors of the Doc2vec simply introduced an additional document vector along with word vectors into Word2vec [19]. Therefore, while training the word vectors, the document vector is trained as well, that gives us the numeric representation of the document. Similar to Word2vec, Doc2vec has two main models which are Distributed Memory (DM) and Distributed Bag of Words (DBOW). DM is analogous to CBOW that uses document feature vector in addition to surrounding words to predict the target word. On the other hand, DBOW is similar to skip-gram that tries to predict randomly sampled words from the paragraph as outputs.

2.5 State-of-the-art language models and representations

By combining the latest achievements in language modelling by means of transformers based on self-attention with the idea of deep contextualized word-piece embeddings together with pretraining universal language model, several NLP and AI research groups introduced universal language models that can be subsequently fine-tuned for a specific NLP task.

Google AI group introduced the so-called bidirectional encoder representations from transformers or BERT for short [20]. Google has made BERT code and implementation available, as well as pre trained BERT models on different languages on huge amounts of data where only minor changes can be done to the model to fine tune it to the tasks needed. On top of this research several frameworks have incorporated the current state-of-the-art models such as DeepPavlov [21], a python library that builds upon BERT, and many others allowing the user to combine them to improve on many NLP tasks.

2.6 Text classification

Text classification is a widely used approach to tackle similar problems such as sentiment analysis and categorization of articles as part of supervised machine learning. Prior to the 1990s, the most common approach were rule-based classification systems, which were manually constructed for each class based on expert opinion [22]. Machine Learning techniques have started to dominate old-fashioned rule-based systems in the following decades, as they help to decrease a remarkable amount of engineering effort on rule construction. When using machine learning models for text classification, there have been many approaches on how to handle the text as input features and how to extract said features from the text. For example, word2vec, TF-IDF vectors, and others aim to map words to a better vector representation in a continuous vector space, thereby improving the performance of the machine learning algorithms.

2.7 Named Entity Recognition

Named Entity Recognition (NER) represents an NLP task that is responsible for extracting the so-called named entities from a text. Named entities may include persons, organizations, locations, time, etc.

The most common classical way of NER is based on CRFs [23].

State-of-the-art methods of NER are also based on the fine-tuning of the pre-trained universal language models such as BERT.

2.8 Knowledge base

Disambiguation frequently implies knowledge about the world than cannot be deduced from the formal text analysis only. To that end various knowledge bases and semantic ontologies may be of use. Some of them aim at the specific lexical areas such as Wordnet [24] that clusters synonyms and allows to identify anotonyms together with a simple hierarchy of hypernyms and hyponyms. Others aim at constructing universal knowledge graphs that represent all the possible knowledge concepts within a single graph with a complex and diverse links between them like Wikidata [25] that stores information from Wikipedia in a structured way available for online querying.

2.9 Chatbots

Chatbot technology was introduced with its first implementation, ELIZA in 1964. It was the first program to make Natural language conversation with a computer possible [26]. ELIZA tackled five problems of a chatbot "the identification of critical words, the discovery of a minimal context, the choice of appropriate transformations, the generation of responses appropriate to the transformation or in the absence of critical words". These are the basic rules still applicable even in modern chatbots.

Today, chatbots have come a long way, where NLP is used in many business typesets for automatic answering and countless other functions. Some of the most used frameworks include Facebook's wit.ai[1] and Google's Dialogflow API[2]. The operation of modern chatbots does not require any stand-alone platforms; they can be integrated into massively used messaging platforms such as Facebook Messenger, Google Assistant or Telegram.

3 Methodology

To answer our research questions, we used data available from SafeCity[3] regarding previous harassment reports in India written by victims in English. Based on this data, we have trained models with different approaches to classify the cases into different kinds of harassment. Then, by using harassment cases correctly identified by the classifier, we aimed to extract spatio-temporal subject information to properly aid the victim. This aid consists of a set of instructions recommended by a chatbot, which is our final product. All the inputs and end products of this project are designed for English texts only.

3.1 Dataset

The SafeCity reports contain around 12,000 precise texts mainly mentioning commenting, ogling and groping issues. Moreover, there are more severe physical harassment cases mentioned as well. Also, it

[1] https://wit.ai/
[2] https://dialogflow.com/
[3] https://safecity.in/

should be underlined that a report naturally may include more than one types of harassment. Figure 1 shows the distribution of several types of harassment in such reports used for this project.

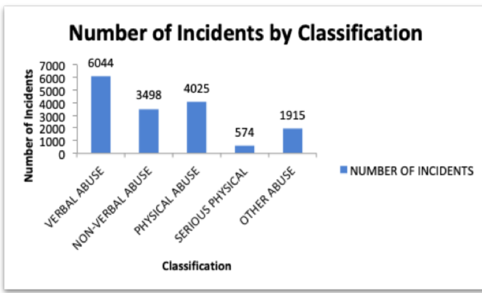

Figure 1: Number of harassment types in SafeCity

3.2 Text pre-processing

Text pre-processing starts with a plain text tokenization that identifies separate tokens that may represent characters, phonemes, morphemes and words. The main difficulty on this stage mainly concerns tokens that cannot be segmented based only on the form of the token, e.g., in case of word tokens the following issues arise: how to handle abbreviations, hyphenated phrases, set phrases written with space between the words, etc.

The next stage includes word lemmatization to obtain word lemmas with morphological analysis in order to acquire word forms and useful grammatical categories such as case, gender and number. This step also involves part-of-speech tagging.

After the tokenization, lemmatization and postagging, the traditional approach is to do syntactic analysis. It may be based on dependencies or on the immediate constituents. This stage allows to extract the necessary information about the word-form relations within each sentence.

Based on the syntactic analysis, various formal semantic parsing techniques are applied.

In this project, we have applied the following preprocessing steps:

- **Contraction handling:** Replacing word contractions such as *I'm* with their unabbreviated form *I am* taking into account misspellings such as *Im*. This was done using regular expressions.
- **Special character removal:** Removing special characters such as $ and double spaces. This was done using regular expressions as well.
- **Spelling correction:** Simple spelling correction function available in Python was added that uses Levenshtein distance [27].
- **Negation handling:** Simple negation handling approach was used in order to identify the word *not* and finding an antonym for the following word, then replacing both not and its following word with the antonym. This was done using the Wordnet synonym-antonym lexicon from the NLTK in Python [24].
- **Lemmatization:** In the feature extraction process for Text Classification models, the corpus was lemmatized in both Bag of Words and Embeddings approaches. This was done using the SpaCy [28].
- **Lower case:** For the majority of tasks (except Named-entity Recognition) the text was converted to lowercase, since this reduced the corpus size and made no difference in most of the tasks.
- **Part-of-Speech Tags:** We used SpaCy again to find out the most frequent POS tags to visualize our reports (See word clouds in the Appendix). Additionally, we created some models using only these tags but dropped this idea since we couldn't observe performance improvements.

3.3 Text classification

In this part of the project, our main goal is to determine whether a report is related to a harassment issue. After that, we want to extract more details about the issue namely types of the harassment or missing information such as time and location in order to suggest proper actions. This would be helpful for our chatbot, in advising convenient actions to different levels of harassment based on the severity of the case such as recommending psychological or medical support.

The initial step was feature engineering where we transform pre-processed text data into feature vectors. There are several approaches in the current state-of-the-art. The first one is a basic counting terms approach that contains part of the text in rows and frequency of each possible term in columns. Since this approach would result in higher values for more repetitive words and longer texts, advanced techniques to find out relative importance of a term were derived such as TF-IDF vectors [29]. These vectors consist of two terms; the first one is Term Frequency (TF), which is the ratio of a specific term in a document. The second one is Inverse Document Frequency (IDF) that is equal to the logarithm of the ratio of the total number of documents over the number of documents containing such term within the corpus. Those vectors can be created based on various input types such as words, characters or combination of N terms (N-grams) [30].

Another approach refers to text embeddings which aim to transform the text into a numerical dense vector representation. It learns the position of a word in high dimension space by using former and latter words in the text. In particular, we used Doc2vec, a special version of Word2vec for documents/paragraphs [19].

After that, Logistic Regression and SVM models were built by using those variables and their performances will be discussed in the results under section 4.

Figure 2 shows a graphical representation of the workflow used to do the classification task.

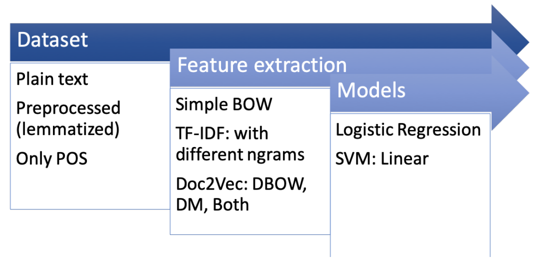

Figure 2: Classification flow

3.4 Named Entity Recognition

To provide specific help to victims of harassment, we are interested in the spatial and temporal information of an incident. Spatial information in this context is the place the harassment has occurred. Temporal information, on the other hand, is information about the date and time of the incident. This information can help us provide the right instructions on which actions a victim should take. To receive these types of information we applied different named entity recognition techniques.

In our project we applied both CRF-based and also state-of-the-art named entity recognition. For the first part we used ready-made solution available in several software packages that are freely distributed, namely:

- Natural Language Toolkit (NLTK) (Python) ("Natural Language Toolkit — NLTK 3.4 documentation," n.d.)
- spaCy library (Python) ("spaCy · Industrial-strength Natural Language Processing in Python," n.d.)
- Stanford CoreNLP software (Java) (Manning et al., 2014)

Each of the package exploits different approach in identifying named entities, so we estimated their capabilities (e.g., entities they are able to extract, the annotation type BIO or BILUO) and identified the one that best fits our needs.

We considered entities only on the sentence level. So no co-reference and anaphora resolution was applied.

For the state-of-the-art approach, we considered Google AI BERT encapsulated into DeepPavlov framework and which was fine-tuned on several widely-used NER datasets for benchmarking such as OntoNotes [31] and CoNLL 2003 [32].

3.5 Fine-tuning

Last year gave a rise to new successfully applied trends in NLP, namely, unsupervised universal language model pretraining and a subsequent fine-tuning of such a model to the specific NLP task. Fine-tuning is basically a transfer learning. It is applied while using a pre-trained generative language model [14], [20]. Large neural networks have been trained on general tasks such as language modelling and then fine-tuned for classification tasks. Particularly, NLP tasks can be fine-tuned with the same single model. In our project we used approach based on universal language model fine-tuning for named entity recognition, namely, pre-trained BERT model for NER task was used for sequence tagging. The framework that is used for Bert NER is DeepPavlov[4]. The model is based on the Transformer architecture [9].

3.6 Knowledge base

The problem of the wrong NER labeling for cases when a location is labelled as a a person has been addressed by means of Wikidata. Namely, each person entity is being queried and checked for the presence of the property related to geographical coordinates in the knowledge base. If such property is found there then the person tag is relabelled to the location.

3.7 Chatbot

A chatbot is a program that can recognize the language and queries of the user and reply with appropriate responses. It is important to design an intuitive architecture for conversational user experience.

3.7.1 Architecture

The conversation must be designed to gather all the data required to provide correct information, an example of incomplete information,

U: Hello

A: Hello, how are you feeling today?

U: Not very well.

A: May I ask what happened?

U: I was walking down the xyz street, where a group of men called me mean things.

A: I'm so sorry that happened to you. I will try my best to help you with this issue.

This dialogue doesn't give detailed information such as the type of the incident, time and location that are required to provide useful information to the user. "mean things" cannot be classified into any kind of legit harassment type. This is why it is important for the chatbot to get direct answers from the user with clear information. To overcome this, current chatbots work on a slot filling based architecture.

3.7.2 Slot based dialogue reading

This is a way to represent the meaning of the sentences. This dialogue system follows "frame and slot" semantics. For example,

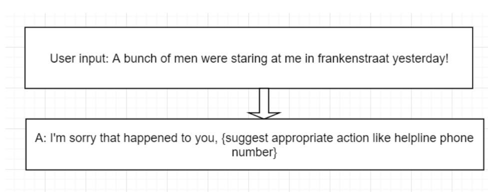

Figure 3: Example of NLU

[4]http://docs.deeppavlov.ai/en/master/components/ner.html

Here, the slots are @time and @location and @date.

@Date : Dates can be given in international formats like mm/dd/yy or verbally written like 24th of april etc..

@Time : Yesterday cannot be a valid time slot, so the system has to reply with an query for asking for exact time. For eg: I'm sorry that happened to you, I am trying to get the help you need, but I need the exact time frame of the incident. Alternatively, we can use the system time to understand the meta like yesterday and today.

@location: Frankenstraat is a valid slot location.

This can also be called as intents and entities. The intents will be the different type of harassment's, and entities are the slots, i.e. date, time and locations.

Intents:

- Physical Abuse
- Verbal Abuse
- Non-Verbal Abuse

3.8 Approaches for chatbot

Nowadays, chatbots can be broadly classified as rule-based (scripted) or Deep learning chatbots. For this project, we tried both but decided to go with a rule-based bot because of the lack of necessity for a deep learning chabot and data for training a dataset being very small for the deep learning to be useful.

3.8.1 Tensorflow[5] chatbot

The use of artificial neural networks to create chatbots is increasingly popular nowadays, however, teaching a computer to have natural conversations is very difficult and often requires large and complicated language models. It requires a very precise and clean dataset with a huge number of cases for the model to be trained to have high precision since we do not possess that we decided not to further approach this tensorflow method [33].

3.8.2 Dialogflow API

Google's dialogflow[6] offers an interactive way to design a chatbot. It solves many complexity in chatbot design but falls down when in need to design a chatbot with relatively large database because classification of intents and entities has to be done manually and not feasible for 10k iterations.

3.8.3 RASA NLU-Core

One of the approaches we tried was RASA[7] framework. It is an open source machine learning framework that can be used to create AI assistants. It has two blocks of models, Natural language understanding (NLU) model and core (NLG+web hookup) model. The NLU model applies understanding the user queries to answer relevant questions or reply based on it being for information or chitchat. The core is for holding conversations and understanding/deciding what to do next. This approach is very flexible for a user and can design a chatbot from scratch and can take forward the approaches in any direction. However, the process is comparatively complicated compared to a rule-based chatbot and also more time-consuming. One more problem of this approach was that the data that is needed to train a model has to be annotated, but the safe city data we are using is either too small or not annotated. Due to the time constraints for this project and lack of proper data set, we decided to go for a semi-preprogrammed chatbot which works on an established platform called telegram.

3.8.4 Telegram API

Telegram[8] is a mass communication application used worldwide similar to alternative applications such as

[5]https://www.tensorflow.org/

[6]https://dialogflow.com

[7]https://rasa.com/

[8]https://telegram.org/

Facebook Messenger or WhatsApp[9]. Telegram has support where users can interact with bots by sending them messages, commands and inline requests. The bot created by the API can be specialized for our use case by integrating our NLP platform for question answering. A script was written based on the intents and entities of the several scenarios with appropriate reply vocabularies using python and Telegram API.

3.9 Chatbot - Dialogue flowchart

The ultimate goal of the chatbot is to provide the user an usable information based on their input. This has to be as diverse as possible and not the make bot very stoic and the conversations must be natural and efficient at the same time. The overall work chat flow is shown in figure figure 4 and each block will have chatbot reply with unique sentences which was framed with the help of Mary Kaltenburg[10].

Initially the Chatbot greets the user and asks for information about the possible harassment event, if the user's input is not classified as a harassment case, the chatbot continues to ask persistently. At every step for the user the text sent by the user is concatenated to its previous inputs and its sent to the classification and named entity recognition system for evaluation. Once the text is classified as harassment, depending on whether the location, date and time information could be retrieved the chatbot either asks the user for that information if it was missing, or asks the user to confirm the retrieved location, date or time information from the previous input. When there is some slot (location, date or time) missing information the chatbot will ask the user for the details up to 3 times per slot and continue asking for information to fill the next slot. Once all the slots are filled or the attempts to do so have been executed, depending on the type of abuse (Physical, Verbal, Non Verbal) identified in the users input the chatbot will provide specific information to the user depending on the case.

When physical abuse is detected, the chatbot asks the user if it needs medical assistance, if the user answers 'yes' it provides contact hours, telephone of the Emergency Department of Maastricht UMC+. If, the user does not need medical assistance, or the information of the Emergency Department of Maastricht UMC+ has already been provided, then the chatbot gives information related to physical abuse and entities with location, hours, and telephones which the user can contact to get aid, such as: Centrum Seksueel Geweld Limburg (CSG Limburg), Acute care (for crises or emergencies), GGD Zuid Limburg-Centrum voor Seksuele Gezondheid (Burgers). When Verbal abuse is detected, the chatbot does not provide information about medical assistance but instead, gives the user contact hours and information of fier.nl [11] an online chat for support for this kind of abuse. When non verbal abuse is detected, the chatbot does not provide information about medical assistance but instead, gives the user contact hours and information of Against her will, another organization specialised in this kind of abuse.

After the different information regarding the type of abused detected in the user's text has been provided, the user is asked if it has reported the event to the police, in case the user answers 'no', information about the police with contact hours and telephone is provided. If the user answered yes or the police information was provided, the chatbot asks the user if it found the chatbot to be useful, this will help to add the user case as labeled data to the training data of the models in the future provided that the user accepts for the chatbot to keep the data. Then, the chatbot shows all the text sent by the user since the start of the conversation and asks for permission to keep the user's data anonymously for further use. If the user accepts, the data is kept, otherwise it is erased. Finally the chatbot ends the conversation with the user.

[9] https://www.whatsapp.com/

[10] United Nations university - Maastricht

[11] https://www.fier.nl/chat

4 Results

4.1 Classification models

We have 4 classification problems as follows:

- **Harassment or not:** First of all, we wanted to see that at what level we can diversify a harassment case from any similar short text which is written by a user on the Internet. Therefore, we collected datasets consisting of some user reviews on IMDB, Amazon or tweets on Twitter as the negative class of our target.
- **Labeling verbal abuses among all harassment reports:** As a next step, we created models in order to catch verbal abuses among all harassment cases. We already had those labels thanks to the SafeCity dataset.
- **Labeling non-verbal abuses among all harassment reports:** Similar to verbal models.
- **Labeling physical abuses among all harassment reports:** Since the number of serious physical abuses was low, they were merged to physical abuses.

For these models, different datasets were created in which numbers of 1s and 0s were in balance. In order to compare candidate models properly, 30% of the data were selected as a test set which was stratified by the target. Then, combinations of various text types, feature extraction methods and modeling techniques were implemented as can be seen in Figure 2.

In the final models, which are input for the chatbot, two models for each classification problem were created by using pre-processed (lemmatized) text. Those use TF-IDF with up to 3 n-grams and Doc2Vec with Distributed Bag of Words (DBOW) approaches respectively. We decided to use these different approaches since both resulted in a good performance in the test set and ensembling them in the chatbot would give us more robust outcomes.

As the classification model, both use Logistic Regression since it has performed better than SVM and returns the probability that gives us the flexibility to change cutoff. Our chatbot is capable of process incoming texts through the same steps and classify. All results can be seen in the Appendix in detail. However, Figures 4 and 5 show the performance of final models on test sets.

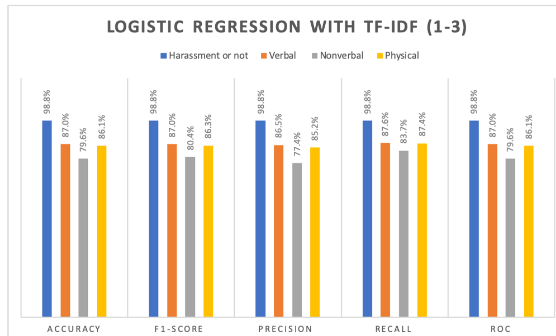

Figure 4: Logistic Regression Models with TF-IDF (up to 3 ngrams)

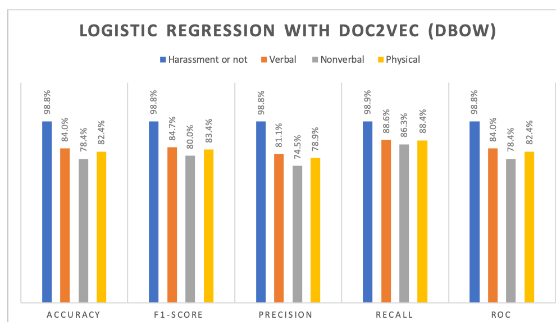

Figure 5: Logistic Regression Models with Doc2Vec (DBOW)

5 Validation

After we were able to show that the classification model delivers acceptable results, we went on to validate the other two parts namely the named entity recognition and the chatbot.

5.1 NER Validation

For the named entity recognition we did a validation using a self made dataset. We created this dataset by writing 5 short reports of harassment cases. In these reports we set the named entities with placeholder variables. To show that the NER model works for a variety of different named entities we downloaded a list of 12900 city names from around the world from SimpleMaps[12]. To further verify that the model is able to identify date and time information in a text we chose different formats to represent those information that can be inserted into the reports. Examples for the date format are: "yesterday", "5 months ago" or "on the 5th July 2019". On the other hand examples for a time format are: "around 10am", "at 10 o'clock" or "at night".

In the next step we inserted these location, date and time information into the reports at the designated positions randomly. Subsequently we put the resulting reports into the different NER models and compared the results provided by these with the original named entities. To avoid cases in which the detected named entities match except for the prefix we removed prefixes from both strings.

Classifier	Location	Date	Time
BERT with Ontonotes	0.92	0.934	0.798
BERT with CoNLL	0.976	-	-
Stanford	0.45	0.2	0.1

Table 1: Validation results

To receive comparable results for the three different used NER models namely Stanford, BERT trained on CoNLL corpora and BERT trained on Ontonotes we generated for each report template 100 variations with randomly picked named entities and used them as input for the models. Table 1 shows the result of these tests. It can be seen that both BERT models deliver reasonable results for the identification of location entities. However the BERT model trained on CoNLL corpora is not able to identify any information about the date or time. However the results produced by BERT are significantly better than the results from the Stanford NER model. The drop of accuracy for time information in the BERT model can be explained by looking at the returned values. As it seems like there can occur some confusion between date and time information.

[12]https://simplemaps.com/data/world-cities

5.2 Chatbot Validation

Because of the complexity of the chatbot dialogue flow we were not able to validate the chatbot entirely. Thus we decided to write scripts for some specific showcases and compare the responses given by the chatbot with the responses we expected.

In the first scenario we don't greet the chatbot at all and just report to it an incidence that is clearly a form of physical harassment. We also provide all necessary information about the location, date and time of the incident directly in the first message. Thus the bot just asks us to confirm this information. In the next step we expect that the bot asks if we need medical assistance. We decline that and the bot gives us the contact details of CSG Limburg, acute care and the GGD Zuid Limburg-Centrum voor Seksuele Gezondheid. Afterwards the bot asks us if we reported the incidence to the police. We answer with yes, so the bot does not give us any additional information and just asks us if it was helpful. To try out if everything is working we answer with no. In the last step the bot asks if it can store the data anonymously. We accept this and the bot ends the conversation as expected.

In the second scenario we greet the bot and introduce ourselves as John in the first message. Thus we expect the bot to ask us about the incident. So the second message we send describes an incident that can be categorized as a form of verbal abuse. But this time we do not provide any information about the location, date or time at all. So we expect the bot to ask us about the location this incident took place. So we tell the bot that this took place "in Maastricht" and confirm with yes after the it asks us if this is correct. In the next step the bot asks us about the date on which the abuse occurred. Again we give it the answer straight away by replying with "yesterday" and confirm with yes. Lastly the bot asks us at which time it occurred and we answer with "at 10am" and confirm once again. In the next step, since the report clearly described an incident of verbal abuse the bot gives us the contact information of fier.nl and asks us if the police was already informed. We reply with "no" and receive the contact information of the local police department. Afterwards the bot asks us again if it was helpful. We answer with yes this time and the bot then asks for permission to store our data. This time we refuse and the bot says us goodbye and ends the conversation.

In the last scenario we send the bot a message that clearly has nothing to do with any form of sexual harassment. Hence we expect the bot to ask for more information. So in the next message we report an incidence that falls under the category of non-verbal abuse. But again we do not provide any information about the location, date or time. Thus the bot asks us where and when this took place. We reply three times with a message that clearly does not contain any information about the location, date or time. Thus the bot continues by giving us information about "Against her will" and asking us if it was reported to the police, if the bot was helpful and if it can store the data.

The complete transcripts of the conversation can be found in the appendix in figure 15, figure 16 and 17. The responses of the bot match the chat dialogue flow described in section 3.9. We chose three examples that cover most of the possible flows. Because the results are compatible with the expected outcome we can conclude that the implementation of the possible paths is correct.

6 Conclusion

#MeToo is a social movement that has attracted great media attention in recent years, especially in social networks. As global awareness is rising, the goal of the project #MeTooMaastricht was to provide victims in Maastricht with a platform to describe their experiences with sexual harassment. For this purpose, we implemented a chatbot using the Telegram API which is able to refer victims to official institutions that can provide further assistance.

In order to provide the best possible help, we have taken into account various factors relating to the incident, such as the type of harassment that was experienced as well as the location, date and time of the incident. Extracting this information in the most natural way presented a challenge, as it was not trivial to extract from a chat text which form of sexual harassment the victim was experiencing.

To overcome this problem, we began to develop a classification model for various types of sexual harassment. To learn the model, we used a dataset provided by the Indian organization SafeCity, which consists of short reports of sexual harassment incidents. Since the texts in this dataset are not formatted optimally for learning a natural language processing model we also added a text pre-processing step consisting of multiple consecutive steps to facilitate the processing of the input data. Afterwards we extended the dataset with unrelated texts, so that the model would be able to recognize cases in which no sexual harassment took place. Subsequently we tried out different text classification techniques to receive the best results. In the end we decided to use a combination of TF-IDF with up to 3 n-grams and Doc2Vec with Distributed Bag of Words (DBOW) and a Logistic Regression classifier as they provided the best results.

In addition to the classification, we also wanted to achieve the most natural possible determination of the place as well as the date and time of the incident. To achieve this, we used names entity recognition. Since there are multiple software packages available for this task, we tried out the most common ones and compared the results. In the end we were satisfied with the results of the state-of-the-art approach BERT.

In the end we used the Telegram Bot API in Python to provide a slot based chatbot that processes a dialogue flow proposed by Mary Kaltenburg and uses the two previously described workflows to extract the necessary information.

References

[1] Sweta Karlekar and Mohit Bansal. Safecity: Understanding diverse forms of sexual harassment personal stories. In *EMNLP*, 2018.

[2] R. L. Binder. Why women don't report sexual assault. *The Journal of Clinical Psychiatry*, (42):437–438, 1981.

[3] C. Y. Suen. n-gram statistics for natural language understanding and text processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):164–172, April 1979.

[4] Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048, 2010.

[5] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014.

[6] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[7] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. *CoRR*, abs/1601.06733, 2016.

[8] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. *CoRR*, abs/1705.04304, 2017.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[10] Brian Roark, Murat Saraclar, and Michael Collins. Discriminative n-gram language modeling. *Comput. Speech Lang.*, 21(2):373–392, April 2007.

[11] Y. Tachioka and S. Watanabe. Discriminative method for recurrent neural network language models. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5386–5390, April 2015.

[12] Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 188–191, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[13] Miikka Silfverberg, Teemu Ruokolainen, Krister Lindén, and Mikko Kurimo. Part-of-speech tagging using conditional random fields: Exploiting sub-label dependencies for improved accuracy. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 259–264. Association for Computational Linguistics, 2014.

[14] Jeremy Howard and Sebastian Ruder. Fine-tuned language models for text classification. *CoRR*, abs/1801.06146, 2018.

[15] Andrew M. Dai and Quoc V. Le. Semi-supervised sequence learning. *CoRR*, abs/1511.01432, 2015.

[16] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[17] Xin Rong. word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*, 2014.

[18] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick,

Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016.

[19] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196, 2014.

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[21] Mikhail Burtsev, Alexander Seliverstov, Rafael Airapetyan, Mikhail Arkhipov, Dilyara Baymurzina, Nickolay Bushkov, Olga Gureenkova, Taras Khakhulin, Yuri Kuratov, Denis Kuznetsov, Alexey Litinsky, Varvara Logacheva, Alexey Lymar, Valentin Malykh, Maxim Petrov, Vadim Polulyakh, Leonid Pugachev, Alexey Sorokin, Maria Vikhreva, and Marat Zaynutdinov. DeepPavlov: Open-source library for dialogue systems. In *Proceedings of ACL 2018, System Demonstrations*, pages 122–127, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[22] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.

[23] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[24] George Miller. *WordNet: An electronic lexical database.* MIT press, 1998.

[25] Denny Vrandečić and Markus Krötzsch. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, September 2014.

[26] Joseph Weizenbaum. Computer power and human reason: From judgment to calculation. 1976.

[27] Peter Norvig. How to write a spelling corrector. *Online at: http://norvig. com/spell-correct. html*, 2007.

[28] Matthew Honnibal and Ines Montani. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 2017.

[29] Wen Zhang, Taketoshi Yoshida, and Xijin Tang. A comparative study of tf* idf, lsi and multi-words for text classification. *Expert Systems with Applications*, 38(3):2758–2765, 2011.

[30] William B Cavnar, John M Trenkle, et al. N-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval*, volume 161175. Citeseer, 1994.

[31] Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. Ontonotes: The 90 In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, NAACL-Short '06, pages 57–60, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[32] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 142–147, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[33] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems*

Design and Implementation ({OSDI} 16), pages 265–283, 2016.

7 Appendix

Figure 6: Word-cloud of nouns in harassment reports

Figure 7: Word-cloud of verbs in harassment reports

Figure 8: Word-cloud of adjectives in harassment reports

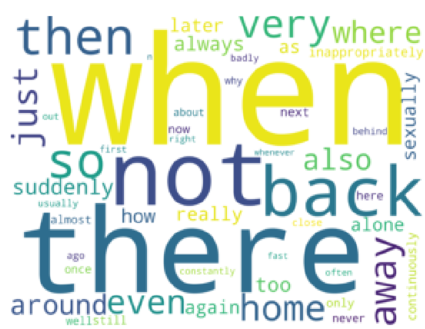

Figure 9: Word-cloud of adverbs in harassment reports

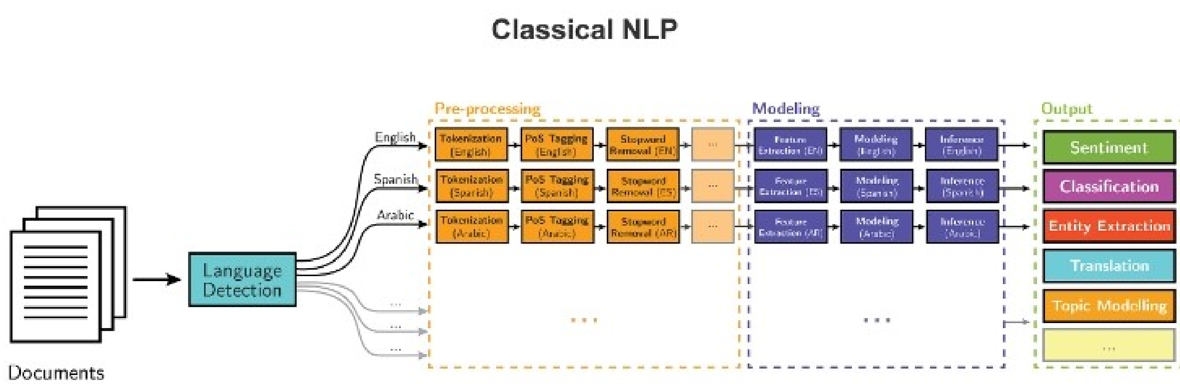

Figure 10: NLP pipelines comparison

Harassment or not

	Accuracy LR:	Accuracy SVM:	F1-score LR:	F1-score SVM:	Precision LR:	Precision SVM:	ROC LR:	ROC SVM:	Recall LR:	Recall SVM:
bow	99.0%	87.4%	99.0%	88.8%	98.9%	80.2%	99.0%	87.4%	99.0%	99.3%
tfidf	99.0%	50.0%	99.0%	0.0%	99.0%	0.0%	99.0%	50.0%	99.0%	0.0%
tfidf_3grams	98.8%	85.5%	98.8%	87.2%	98.8%	78.1%	98.8%	85.5%	98.8%	98.6%
doc2vec_dbow	98.8%	98.7%	98.8%	98.7%	98.8%	98.6%	98.8%	98.7%	98.9%	98.9%
doc2vec_dm	73.0%	72.1%	76.4%	77.9%	67.9%	64.4%	73.0%	72.1%	87.4%	98.7%
doc2vec_dbowdb	97.6%	97.7%	97.6%	97.8%	95.9%	95.9%	97.6%	97.7%	99.3%	99.7%

Verbal

	Accuracy LR:	Accuracy SVM:	F1-score LR:	F1-score SVM:	Precision LR:	Precision SVM:	ROC LR:	ROC SVM:	Recall LR:	Recall SVM:
bow	86.9%	76.7%	86.9%	81.0%	86.7%	68.4%	86.9%	76.7%	87.2%	99.3%
tfidf	87.0%	79.4%	87.2%	82.4%	86.1%	72.0%	87.0%	79.4%	88.2%	96.4%
tfidf_3grams	87.0%	75.8%	87.0%	80.4%	86.5%	67.6%	87.0%	75.8%	87.6%	99.2%
doc2vec_dbow	84.0%	83.6%	84.7%	85.6%	81.1%	76.4%	84.0%	83.6%	88.6%	97.5%
doc2vec_dm	64.7%	66.6%	69.1%	74.4%	61.5%	60.3%	64.7%	66.6%	78.7%	97.1%
doc2vec_dbowdb	82.4%	82.9%	84.1%	85.3%	76.7%	74.7%	82.4%	82.9%	92.9%	99.5%

Nonverbal

	Accuracy LR:	Accuracy SVM:	F1-score LR:	F1-score SVM:	Precision LR:	Precision SVM:	ROC LR:	ROC SVM:	Recall LR:	Recall SVM:
bow	78.9%	71.5%	79.5%	77.7%	77.4%	63.7%	78.9%	71.5%	81.8%	99.6%
tfidf	79.4%	73.1%	80.6%	78.5%	76.2%	65.4%	79.4%	73.1%	85.5%	98.3%
tfidf_3grams	79.6%	70.6%	80.4%	77.1%	77.4%	63.1%	79.6%	70.6%	83.7%	99.1%
doc2vec_dbow	78.4%	78.7%	80.0%	82.2%	74.5%	70.7%	78.4%	78.7%	86.3%	98.2%
doc2vec_dm	56.4%	57.2%	61.8%	69.3%	55.0%	54.1%	56.4%	57.2%	70.7%	96.5%
doc2vec_dbowdb	76.8%	74.1%	79.8%	79.4%	70.7%	65.9%	76.8%	74.1%	91.6%	100.0%

Physical

	Accuracy LR:	Accuracy SVM:	F1-score LR:	F1-score SVM:	Precision LR:	Precision SVM:	ROC LR:	ROC SVM:	Recall LR:	Recall SVM:
bow	85.8%	72.6%	85.9%	78.3%	85.3%	64.9%	85.8%	72.6%	86.4%	98.7%
tfidf	85.4%	50.0%	85.8%	0.0%	83.6%	0.0%	85.4%	50.0%	88.2%	0.0%
tfidf_3grams	86.1%	71.8%	86.3%	77.7%	85.2%	64.2%	86.1%	71.8%	87.4%	98.2%
doc2vec_dbow	82.4%	80.6%	83.4%	83.5%	78.9%	72.6%	82.4%	80.6%	88.4%	98.2%
doc2vec_dm	62.6%	62.3%	67.9%	71.8%	59.5%	57.3%	62.6%	62.2%	78.9%	95.9%
doc2vec_dbowdb	81.9%	78.3%	83.9%	82.1%	75.6%	69.7%	81.9%	78.2%	94.4%	99.8%

Figure 11: Classification results with pre-processed text

Harassment or not

	Accuracy LR:	Accuracy SVM:	F1-score LR:	F1-score SVM:	Precision LR:	Precision SVM:	ROC LR:	ROC SVM:	Recall LR:	Recall SVM:
bow	99.1%	87.3%	99.1%	88.7%	99.1%	80.0%	99.1%	87.3%	99.2%	99.4%
tfidf	99.1%	50.0%	99.1%	0.0%	98.9%	0.0%	99.1%	50.0%	99.3%	0.0%
tfidf_3grams	99.1%	85.0%	99.1%	86.9%	99.3%	77.3%	99.1%	85.0%	99.0%	99.1%
doc2vec_dbow	98.8%	98.8%	98.8%	98.8%	98.9%	98.7%	98.8%	98.8%	98.8%	99.0%
doc2vec_dm	81.7%	82.0%	83.7%	84.7%	75.4%	73.8%	81.7%	82.0%	94.1%	99.5%
doc2vec_dbowdb	97.3%	98.0%	97.3%	98.0%	95.4%	96.5%	97.3%	98.0%	99.4%	99.5%

Verbal

	Accuracy LR:	Accuracy SVM:	F1-score LR:	F1-score SVM:	Precision LR:	Precision SVM:	ROC LR:	ROC SVM:	Recall LR:	Recall SVM:
bow	85.4%	75.0%	85.6%	79.9%	84.0%	66.9%	85.4%	75.0%	87.3%	99.2%
tfidf	85.3%	76.0%	86.0%	80.5%	82.1%	67.7%	85.3%	76.0%	90.4%	99.3%
tfidf_3grams	84.6%	73.7%	85.0%	79.0%	82.8%	65.8%	84.6%	73.7%	87.3%	98.9%
doc2vec_dbow	81.8%	82.0%	83.0%	84.5%	77.9%	74.1%	81.8%	82.0%	88.8%	98.4%
doc2vec_dm	62.6%	65.0%	68.1%	73.4%	59.4%	59.2%	62.6%	65.0%	79.9%	96.4%
doc2vec_dbowdb	81.0%	80.9%	83.5%	83.9%	73.8%	72.5%	81.0%	80.9%	96.0%	99.7%

Nonverbal

	Accuracy LR:	Accuracy SVM:	F1-score LR:	F1-score SVM:	Precision LR:	Precision SVM:	ROC LR:	ROC SVM:	Recall LR:	Recall SVM:
bow	78.7%	71.0%	79.2%	77.3%	77.4%	63.4%	78.7%	71.0%	81.0%	99.0%
tfidf	80.3%	50.0%	81.5%	0.0%	76.9%	0.0%	80.3%	50.0%	86.6%	0.0%
tfidf_3grams	79.6%	69.9%	80.2%	76.6%	78.1%	62.6%	79.6%	69.9%	82.4%	98.5%
doc2vec_dbow	78.1%	78.5%	79.6%	81.9%	74.5%	70.6%	78.1%	78.4%	85.5%	97.6%
doc2vec_dm	56.4%	60.4%	61.2%	70.8%	55.1%	56.1%	56.4%	60.4%	68.9%	96.0%
doc2vec_dbowdb	78.4%	75.9%	81.2%	80.6%	71.8%	67.5%	78.4%	75.9%	93.6%	99.9%

Physical

	Accuracy LR:	Accuracy SVM:	F1-score LR:	F1-score SVM:	Precision LR:	Precision SVM:	ROC LR:	ROC SVM:	Recall LR:	Recall SVM:
bow	86.3%	73.2%	86.4%	78.8%	85.8%	65.3%	86.3%	73.2%	87.0%	99.3%
tfidf	86.1%	74.1%	86.6%	79.4%	83.2%	66.0%	86.1%	74.1%	90.4%	99.5%
tfidf_3grams	85.7%	71.8%	85.8%	77.9%	84.8%	64.1%	85.7%	71.8%	86.9%	99.1%
doc2vec_dbow	82.4%	81.1%	83.5%	83.8%	78.7%	73.4%	82.4%	81.1%	89.0%	97.6%
doc2vec_dm	59.7%	58.5%	66.8%	70.3%	56.8%	54.7%	59.7%	58.5%	81.1%	98.1%
doc2vec_dbowdb	79.9%	79.5%	82.4%	83.0%	73.4%	70.9%	79.9%	79.5%	94.0%	99.9%

Figure 12: Classification results with plain text

Harassment or not

	Accuracy LR:	Accuracy SVM:	F1-score LR:	F1-score SVM:	Precision LR:	Precision SVM:	ROC LR:	ROC SVM:	Recall LR:	Recall SVM:
bow	99.0%	86.6%	99.0%	88.1%	98.9%	79.0%	99.0%	86.6%	99.1%	99.5%
tfidf	99.2%	50.0%	99.2%	0.0%	99.3%	0.0%	99.2%	50.0%	99.2%	0.0%
tfidf_3grams	99.1%	83.7%	99.1%	85.9%	99.0%	75.7%	99.1%	83.7%	99.1%	99.2%
doc2vec_dbow	98.5%	98.5%	98.6%	98.5%	98.3%	98.1%	98.5%	98.5%	98.8%	98.9%
doc2vec_dm	81.1%	79.4%	83.6%	82.9%	73.9%	70.9%	81.1%	79.4%	96.2%	99.9%
doc2vec_dbowdb	96.6%	97.2%	96.7%	97.3%	94.3%	94.9%	96.6%	97.2%	99.2%	99.7%

Verbal

	Accuracy LR:	Accuracy SVM:	F1-score LR:	F1-score SVM:	Precision LR:	Precision SVM:	ROC LR:	ROC SVM:	Recall LR:	Recall SVM:
bow	85.4%	74.1%	85.6%	79.3%	84.5%	66.0%	85.4%	74.1%	86.8%	99.1%
tfidf	85.5%	80.9%	86.0%	81.3%	83.5%	79.7%	85.5%	80.9%	88.6%	83.0%
tfidf_3grams	85.6%	72.8%	85.8%	78.5%	84.7%	65.0%	85.6%	72.8%	87.0%	99.0%
doc2vec_dbow	82.6%	81.9%	83.5%	84.5%	79.5%	73.7%	82.6%	81.9%	87.8%	99.0%
doc2vec_dm	69.5%	52.8%	74.1%	67.9%	64.4%	51.5%	69.5%	52.8%	87.4%	99.9%
doc2vec_dbowdb	80.4%	76.6%	82.6%	81.1%	74.3%	68.1%	80.4%	76.6%	92.9%	100.0%

Nonverbal

	Accuracy LR:	Accuracy SVM:	F1-score LR:	F1-score SVM:	Precision LR:	Precision SVM:	ROC LR:	ROC SVM:	Recall LR:	Recall SVM:
bow	78.2%	70.1%	78.7%	76.8%	77.1%	62.8%	78.2%	70.1%	80.3%	99.0%
tfidf	79.9%	50.0%	80.8%	0.0%	77.5%	0.0%	79.9%	50.0%	84.3%	0.0%
tfidf_3grams	78.7%	69.2%	79.1%	76.3%	77.6%	62.1%	78.7%	69.2%	80.7%	98.9%
doc2vec_dbow	77.5%	78.2%	79.0%	81.9%	74.0%	70.0%	77.5%	78.2%	84.7%	98.7%
doc2vec_dm	59.8%	63.2%	66.6%	72.6%	57.0%	57.8%	59.8%	63.2%	80.0%	97.3%
doc2vec_dbowdb	72.6%	72.8%	76.4%	78.6%	67.0%	64.8%	72.5%	72.8%	88.8%	100.0%

Physical

	Accuracy LR:	Accuracy SVM:	F1-score LR:	F1-score SVM:	Precision LR:	Precision SVM:	ROC LR:	ROC SVM:	Recall LR:	Recall SVM:
bow	86.6%	72.5%	86.5%	78.3%	86.8%	64.6%	86.6%	72.5%	86.1%	99.4%
tfidf	86.8%	81.2%	87.1%	82.5%	85.1%	77.4%	86.8%	81.2%	89.3%	88.2%
tfidf_3grams	86.7%	71.7%	86.8%	77.8%	86.0%	64.0%	86.7%	71.7%	87.6%	99.3%
doc2vec_dbow	82.9%	81.0%	83.8%	83.8%	79.6%	73.0%	82.9%	81.0%	88.5%	98.5%
doc2vec_dm	65.5%	57.3%	70.8%	69.9%	61.5%	54.0%	65.5%	57.3%	83.4%	99.3%
doc2vec_dbowdb	77.9%	75.3%	80.6%	80.2%	72.0%	67.0%	77.9%	75.3%	91.4%	99.9%

Figure 13: Classification results with POS text

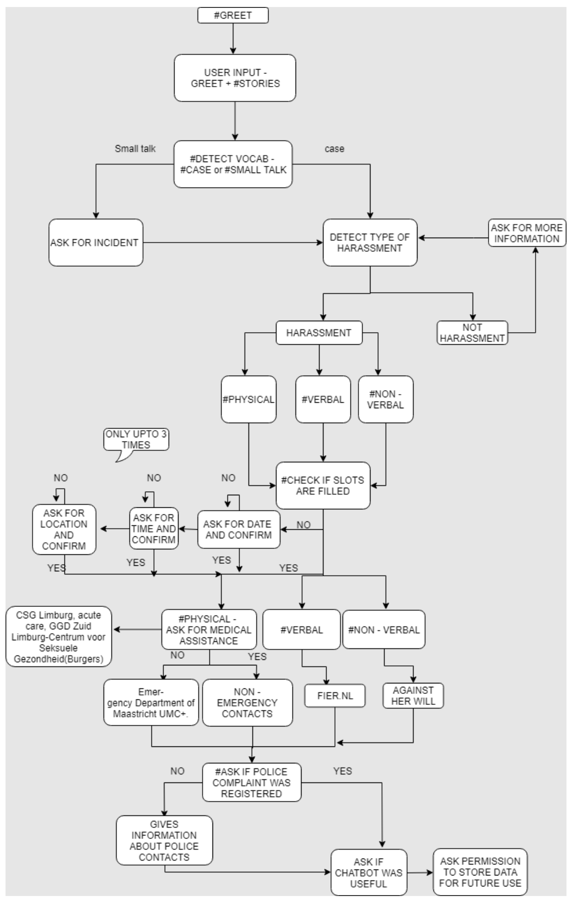

Figure 14: Chat dialogue flow

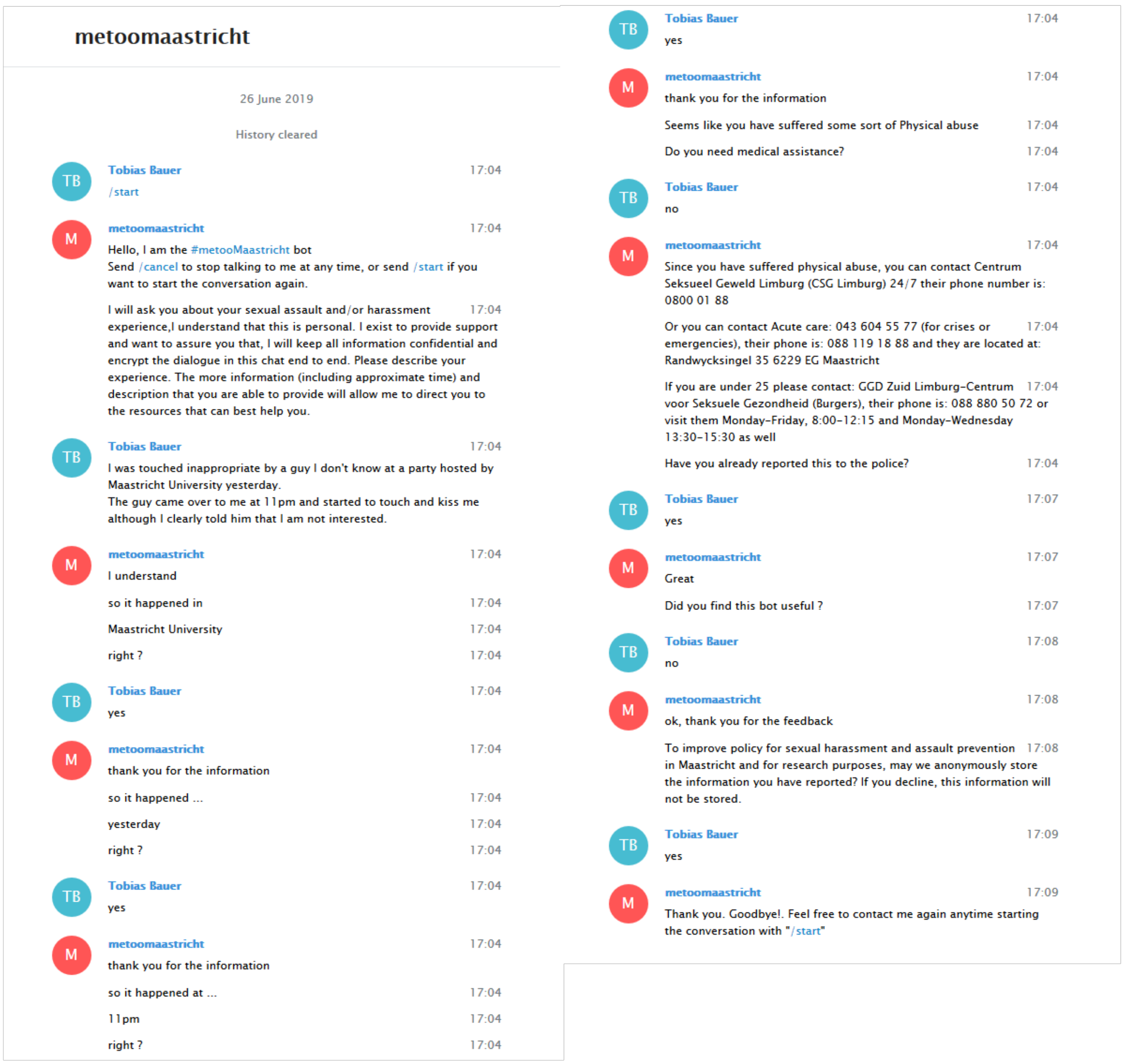

Figure 15: Transcript of scenario 1

metoomaastricht

26 June 2019

History cleared

Tobias Bauer 17:34
/start

metoomaastricht 17:34
Hello, I am the #metooMaastricht bot
Send /cancel to stop talking to me at any time, or send /start if you want to start the conversation again.

I will ask you about your sexual assault and/or harassment experience,I understand that this is personal. I exist to provide support and want to assure you that, I will keep all information confidential and encrypt the dialogue in this chat end to end. Please describe your experience. The more information (including approximate time) and description that you are able to provide will allow me to direct you to the resources that can best help you. 17:34

Tobias Bauer 17:35
Hi I'm John

metoomaastricht 17:35
I understand, can you give me more information please?

Tobias Bauer 17:35
A man came over to me and told me that I look really hot and that he would like to take me home and do some nasty stuff with me.
I told him that I have no interest in going home with him so he started to insult me.

metoomaastricht 17:35
I understand

Please indicate where this experience took place. This does not need to be precise. 17:35

Tobias Bauer 17:35
This took place in Maastricht

metoomaastricht 17:35
so it happened in

Maastricht 17:35

right ? 17:35

Tobias Bauer 17:36
yes

metoomaastricht 17:36
thank you for the information

on what date did it happen? 17:36

Tobias Bauer 17:36
yesterday

metoomaastricht 17:36
so it happened in

yesterday 17:36

right ? 17:36

Tobias Bauer 17:36
yes

metoomaastricht 17:36
thank you for the information

at what time of the day did it happen? 17:36

Tobias Bauer 17:36
at 10am

metoomaastricht 17:36
so it happened at

10am 17:36

right ? 17:36

Tobias Bauer 17:36
yes

metoomaastricht 17:36
thank you for the information

Seems like you have suffered some sort of Verbal abuse 17:36

You can chat online about your experience with fier.nl (dutch only) at https://www.fier.nl/chat Monday to Friday from 7 pm to 6 am Saturday and Sunday from 8 pm to 6 am Holidays from 8 pm to 6 am, or call them 24/7 at 088 - 20 80 000 17:36

Have you already reported this to the police? 17:36

Tobias Bauer 17:37
no

metoomaastricht 17:37
Please report this event to the Police at this phone number: 0900 88 44

And have this in mind when you report this event:
https://www.politie.nl/themas/seksueel-misbruik.html 17:37

Did you find this bot useful ? 17:37

Tobias Bauer 17:37
yes

metoomaastricht 17:37
xxxxxx

good 17:37

This is the info I gathered from you: 17:37
text - Hi I'm John A man came over to me and told me that I look really hot and that he would like to take me home and do some nasty stuff with me.
I told him that I have no interest in going home with him so he started to insult me. This took place in Maastricht yesterday at 10am
Location - Maastricht
Date - yesterday
Time - 10am
Physical - 0
Verbal - 2
NonVerbal - 0
cdate - 3
clocation - 2
ctime - 2

To improve policy for sexual harassment and assault prevention in Maastricht and for research purposes, may we anonymously store the information you have reported? If you decline, this information will not be stored. 17:37

Tobias Bauer 17:37
no

metoomaastricht 17:37
Thank you. Goodbye!. Feel free to contact me again anytime starting the conversation with "/start"

Figure 16: Transcript of scenario 2

metoomaastricht

26 June 2019

History cleared

Tobias Bauer 18:04
/start

metoomaastricht 18:04
Hello, I am the #metooMaastricht bot
Send /cancel to stop talking to me at any time, or send /start if you want to start the conversation again.

I will ask you about your sexual assault and/or harassment experience,I understand that this is personal. I exist to provide support and want to assure you that, I will keep all information confidential and encrypt the dialogue in this chat end to end. Please describe your experience. The more information (including approximate time) and description that you are able to provide will allow me to direct you to the resources that can best help you. 18:04

Tobias Bauer 18:04
Hi I'm just checking this bot

metoomaastricht 18:04
I understand, can you give me more information please?

Tobias Bauer 18:05
This guy I know always looks at me like he's imagining me naked.

metoomaastricht 18:05
I understand

Please indicate where this experience took place. This does not need to be precise. 18:05

Tobias Bauer 18:05
How does this bot work?

metoomaastricht 18:05
Can you be more explicit please?

Tobias Bauer 18:05
I just want to find out what this bot does.

metoomaastricht 18:05
Can you be more explicit please?

Tobias Bauer 18:05
The story I told was just made up.

metoomaastricht 18:05
thank you for the information

on what date did it happen? 18:05

Tobias Bauer 18:07
I don't know

metoomaastricht 18:07
thank you for the information

at what time of the day did it happen? 18:07

Tobias Bauer 18:07
I don't know

metoomaastricht 18:07
Can you be more explicit please?

Tobias Bauer 18:07
No

metoomaastricht 18:07
Can you be more explicit please?

Tobias Bauer 18:07
No

metoomaastricht 18:07
Seems like you have suffered some sort of Non Verbal abuse

you could contact "Against her will" and talk anonymously their phone is: 0592 - 34 74 44, or visit them Monday through Thursday from 2:00 pm to 5:00 pm and from 6:00 pm to 18:07

Phone: 0592 - 34 74 44 18:07

Hours: Monday through Thursday from 2:00 pm to 5:00 pm and from 6:00 pm to 9:00 pm 18:07

Have you already reported this to the police? 18:07

Tobias Bauer 18:08
yes

metoomaastricht 18:08
Great

Did you find this bot useful ? 18:08

Tobias Bauer 18:08
no

metoomaastricht 18:08
ok, thank you for the feedback

To improve policy for sexual harassment and assault prevention in Maastricht and for research purposes, may we anonymously store the information you have reported? If you decline, this information will not be stored. 18:08

Tobias Bauer 18:08
yes

metoomaastricht 18:08
Thank you. Goodbye!. Feel free to contact me again anytime starting the conversation with "/start"

Figure 17: Transcript of scenario 3