University of Ljubljana
Faculty of Computer and Information Science

Drejc Pesjak

# Hate speech paraphraser

bachelor's thesis

university study programme
undergraduate programmes
computer and information science

Mentor: prof. dr. Zoran Bosnić
Comentor: prof. dr. Marko Robnik Šikonja

Ljubljana, 2022

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Drejc Pesjak

# Parafraziranje sovražnega govora

diplomsko delo

univerzitetni študijski program
prve stopnje
računalništvo in informatika

Mentor: prof. dr. Zoran Bosnić
Somentor: prof. dr. Marko Robnik Šikonja

Ljubljana, 2022



*The text is formatted with LaTeX.*

**Candidate:** Drejc Pesjak
**Title:** Hate speech paraphraser
**Type of thesis:** Bachelor thesis on the university study programme first-degree Computer and Information Science
**Mentor:** prof. dr. Zoran Bosnić
**Co-mentor:** prof. dr. Marko Robnik Šikonja

**Description:**
In the thesis, the candidate shall tackle the problem of hateful speech. He shall propose a novel method, which instead of removing hateful comments shall enable their replacement using the paraphrasing techniques from the field of natural language processing. The candidate shall evaluate the method's performance using objective or subjective approaches.

**Naslov:** Parafraziranje sovražnega govora

**Opis:**
Kandidat naj v diplomski nalogi obravnava problematiko sovražnega govora. Predlaga naj metodo, ki naj namesto odstranjevanja sovražnih komentarjev omogoča njihovo zamenjavo s tehnikami parafraziranja s področja metod za obdelavo naravnega jezika. Uspešnost metode naj objektivno ali subjektivno ovrednosti.

# Contents

# List of abbreviations

| Abbreviation | Meaning |
|---|---|
| **HS** | Hate Speech |
| **FoS** | Freedom of Speech |
| **CN** | Counter Narrative |
| **NLP** | Natural Language Processing |
| **NLI** | Natural Language Inference |
| **SVM** | Support Vector Machine |
| **GBDT** | Gradient Boosted Decision Trees |
| **CNN** | Convolutional Neural Network |
| **RNN** | Recurrent Neural Network |
| **BiRNN** | Bidirectional Recurrent Neural Network |
| **LSTM** | Long Short-Term Memory |
| **BLEU** | BiLingual Evaluation Understudy |
| **ROUGE** | Recall-Oriented Understudy for Gisting Evaluation |
| **GAN** | Generative Adversarial Networks |
| **GPT** | Generative Pre-Training |
| **BERT** | Bi-directional Encoder Representations from Transformers |
| **ALBERT** | A Little BERT |
| **T5** | Text-To-Text Transfer Transformer |

# Abstract

**Title:** Hate speech paraphraser

**Author:** Drejc Pesjak

There is plenty of hate speech on the web, which is additionally enabled by the possibility to remain anonymous, and many forums as well as news websites are trying to fight against it with a large number of moderators that remove hateful comments. Due to large numbers of daily comments they use automated hate speech detection software. We propose a DPhate system, which outputs an unhateful alternative to the posted hateful comment. The system uses a series of pre-trained paraphrasing models, that generate non-hateful sentences. The automatic evaluation has shown that in 84.37% of cases at least one acceptable sentence is generated, whereas only 67.90% of rephrasals were deemed acceptable by human evaluators.

**Keywords:** hate speech, natural language processing, transformers, BERT models, machine learning, paraphrasing.

# Povzetek

**Naslov:** Parafraziranje sovražnega govora

**Avtor:** Drejc Pesjak

Splet je poln sovražnega govora, ki ga dodatno spodbuja možnost anonimnosti. Mnogi forumi in novičarske spletne strani se branijo z moderatorji, ki odstranijo škodljive komentarje. Ker je po navadi komentarjev veliko (več deset tisoč na dan), si moderatorji pomagajo s programi za avtomatsko zaznavanje sovražnega govora. V svoji diplomski nalogi predlagamo nov sistem DPhate, ki uporabniku ob objavi sovražnega komentarja predlaga nesovražno alternativo z ohranjenim pomenom. V sistemu uporabimo več prednaučenih modelov, ki s parafraziranjem generirajo nesovražne povedi. Avtomatska evalvacija je pokazala, da se v 84.37% generira vsaj en primeren stavek, medtem ko so generirane parafraze človeški evalvatorji ocenili za primerne v 67.90%.

**Ključne besede:** sovražni govor, obdelava naravnega jezika, transformerji, modeli BERT, strojno učenje, parafraziranje.

# Razširjeni povzetek

Poleg odraslih tudi otroci posegajo po napravah, ki imajo dostop do spleta, in s tem do različnih spletnih vsebin, tako primernih kot neprimernih. Slabih informacij, kot je sovražni govor, se, predvsem zaradi njihove mentalne nezrelosti, ne morejo obraniti in tako srkajo vase tudi neprimerne vsebine. Izpostavljenost negativnim vplivom nanje vpliva zelo negativno [1].

Spletne strani (forumi, razne družabne platforme in novičarske strani) bijejo bitko proti sovražnemu govoru. Ker želijo ustvariti prijazno okolje za njihove uporabnike, morajo postaviti pravila komuniciranja. Takšna pravila so prisotna na vseh večjih platformah, kot so Facebook, Youtube, Twitter, New York Times ali Reddit [2]. Pri zatiranju sovraštva se nagibajo k moderiranju komentarjev (ter drugih uporabniških vsebin), tj. brisanje sovražnih ali žaljivih komentarjev, oz. komentarjev, ki ne upoštevajo smernic skupnosti (angl. community guidelines). "Spletni policaji", ki uveljavljajo ta pravila, so moderatorji. Gre za ljudi, ki jih posamezno podjetje najame, da pregledujejo vsebino na njihovem spletnem mestu, in ukrepajo v primeru kršitve. Ker pa je komentarjev ponekod tudi več deset tisoč na dan, se podjetja dodatno poslužujejo avtomatiziranih sistemov za prepoznavo sovražnega govora v besedilu. Algoritmi niso dovolj natančni, zato morajo označene komentarje pregledati še ljudje.

Za zaznavo sovražnega govora so bili v začetku uporabljeni algoritmi za strojno učenje, kot so odločitvena drevesa ali SVM, ki klasificirajo na podlagi značilk pridobljenih iz besedila. Kasneje je bilo pridobivanje značilk že zajeto v metodah, npr. v nevronskih mrežah prilagojenih za obdelavo besedila, kot

so npr. CNN, RNN in LSTM. Čeprav so te metode vračale dobre rezultate (tudi z F1-oceno do 93% [3]), pa niso primerljive z najsodobnejšimi algoritmi. Dandanes na področju NLP (obdelava naravnega jezika) dominirajo arhitekture, ki temeljijo na arhitekturi transformer [4]. Transformer je sestavljen iz kodirnika in dekodirnika, pri čemer prvi izlušči pomen, drugi pa nato iz njega sestavi novo poved. Model, ki za predstavitev besedila uporablja kodirnik transformerja, je BERT [5]. BERT je najprej učen na veliki množici splošnega besedila in nato prilagojen na manjši množici za specifično nalogo. Takšen način učenja se imenuje učenje s prenosom znanja (angl. transfer learning). Jezikovni model BERT so razvili v podjetju Google, osnovna verzija BERT Base pa ima 12 slojev in 110 milijonov parametrov. Zaradi velikosti modela in velike učne množice pa je pristop zelo učinkovit v večini problemov za obdelavo jezika, tudi za zaznavo sovraženga govora.

Zaznava in odstranitev sovražnega govora lahko krši pravico do izražanja. Gre za nekakšno bitko med sovražnim govorom in svobodo govora. Države to pravico različno urejajo. V Sloveniji je dovoljeno svobodno izražanje, če ne razpihuje sovraštva do neke skupine (kar je kaznivo dejanje [6]). V Združenih državah Amerike je svoboda govora ustavna pravica, sovražni govor pa formalno ne obstaja. Splet, ki ne pozna državnih meja, se nagiba k ameriški interpretaciji, saj naj bi bil splet svoboden, kar omogoča vsakemu posamezniku, ne glede na starost, spol ali raso, da izrazi svoje mnenje, četudi je v nasprotju z mišljenjem večine.

Namesto odstranjevanja sovražnih komentarjev, ki bi omejevalo pravico do svobodnega izražanja, ali opustitve regulacije, kar bi lahko bilo škodljivo za nekatere skupine, v tem delu predlagamo boljšo rešitev. Ob oddaji komentarja na določeni platformi bi v primeru sovražnega sporočila, uporabniku predlagali prijaznejšo različico z enakim pomenom. Podobno idejo so predlagali Salminen in sod. [7], ki so se ukvarjali z odstranitvijo sovražnega dela sporočila. To so dosegli z nevronsko mrežo za zaznavanje sovražnosti, ki je zaznala, katere besede so privedle do takšne odločitve, nato pa so odstranili del odvisnega stavka s temi besedami (odvisnost so določili z drevesnimi

odvisnicami). Pri takšnem odstranjevanju lahko pride do izgube pomena povedi. Zato avtorji predlagajo, da bi bilo potrebno razviti generativen pristop, na kar se osredotočamo v našem delu.

V delu smo uporabilinekatere že obstoječe komponente. Kot osnovo smo vzeli podatkovno zbirko označenega sovražnega govora *Hatexplain* [8], kjer je bilo potrebno namesto značk <user>, <percent>, <number> vstaviti naključne vrednosti, saj jih drugače detektor sovražnega govora ne bi znal interpretirati. Kot poenostavitev smo odstranili tudi smeškote. Vhodne primere smo pognali skozi algoritem *DPhate*, ki najprej razširi skrajšane besedne zveze ("don't", "isn't", "won't" se spremenijo v "do not", "is not", "will not" ), nato pa jim še odstrani sovražne pridevnike in prislove, za katere se je v podatkovni analizi izkazalo, da nosijo najmanj informacije (sovražne besede identificiramo s pomočjo knjižnice *better_profanity*, besedne vrste pa z metodo *pos_tag* iz knjižnice *NLTK*). Iz obdelane povedi nato s parafraziranjem generiramo več podobnih povedi (za to smo uporabili knjižnico *Pegasus* [9], kjer lahko z nastavljanjem parametrov vplivamo na raznolikost generiranih povedi). Nove povedi z detektorjem *Detoxify* [10] preverimo glede sovražnosti, nesovražne pa primerjamo z originalno povedjo glede podobnosti. Podobnost je izračunana s kosinusno podobnostjo na vektorskih vložitvah pridobljenih iz modela BERT [11]. Na koncu še izločimo povedi napisane v celoti z velikimi črkami in tiste, ki vsebujejo artefakte generatorja podobnih povedi. Če metoda ni našla primerne rešitve (ker so vse generirane povedi še vedno sovražne ali pa niso več podobne originalni) se postopek, brez razširjanja in odstranjevanja vulgarnih pridevnikov in prislovov, še enkrat ponovi, a tokrat na najmanj sovražni generirani povedi.

Generirane pare povedi smo uporabili tudi za dodatno učenje jezikovnega modela T5 [12], ki pa ni vrnil dobrih rezultatov, saj povedi še niso bile ocenjene z avtomatskimi merami ali s pomočjo človeških evalvatorjev.

Evalvacija je potekala po dveh kriterijih, sovražnosti generirane povedi in njene podobnosti z originalno, pri čemer je zaželena nizka vrednost prvega in visoka vrednost drugega kriterija. Evalvacije sistema DPhate smo se lotili

z dveh vidikov.

Pri avtomatski evalvaciji smo uporabili vnaprej naučene jezikovne modele. Za evalvacijo sovražnosti smo izkoristili ansambelsko metodo skladanja (angl. "stacking"), kjer so za osnovo vzeti trije vnaprej naučeni modeli za zaznavo sovražnega govora, njihove napovedi pa se posredujejo meta modelu (večrazredna logistična regresija), ki jih združi in vrne končni rezultat. Takšna arhitektura je dosegla napovedno točnost 79.78%. Za evalvacijo podobnosti smo vzeli model simSCE, ki ima povprečno natančnost 83.76% in je v času pisanja najboljši model na podatkovnih zbirkah SICK in STS14. Evalvacija je pokazala, da se je 77.92% parom povedi zmanjšala sovražnost in da je 92% generiranih povedi podobnih originalnim, medtem ko je 70.74% povedi zadostilo obema kriterijema. Ker se za vsako vhodno poved generira več podobnih povedi, je 84.37% možnosti, da bo vsaj ena zadoščala obema kriterijema.

Druga evalvacija je slonela na človeških ocenjevalcih. Na platformi za množičenje MicroWorkers smo objavili kampanjo, na katero se je prijavilo 61 ljudi iz angleško govorečih držav. Ocenili so 876 primerov, pri čemer so vsakega pregledali trije različni evalvatorji. V anketi so kriterije lahko ocenili z oceno od 1 do 5. Človeški evalvatorji so 81.77% primerov ocenili kot nesovražne in 51.26% generiranih povedi kot podobnim originalnim, zgolj 36.35% pa jih je zadoščalo obema kriterijema. Ker pa je vsakič generiranih več odgovorov, je verjetnost, da vsaj eden zadošča obema kriterijema, 67.90%.

Sistem DPhate še zdaleč ni popoln. Težave nastopijo, če vhodno besedilo vsebuje sleng ali napačno črkovane besede. Velikokrat sistem ne najde rešitve za krajše povedi, saj po odstranitvi sovražnega dela ne vsebujejo dovolj informacije, da bi bilo možno tvoriti celotno poved. Takšne povedi so po navadi napisane z namenom, da užalijo naslovnika in ne nosijo nobenega dodatnega pomena. Dodatno težavo predstavlja hitrost. Trenutni algoritem lahko za en primer porabi tudi pol minute. Glavno ozko grlo je model za parafraziranje, kjer je čas generiranja odvisen od dolžine vhodne povedi in

števila generiranih povedi.

Druge mogoče izboljšave so pretvorba slengovskih besed v njihove slovarske sopomenke in popravljanje napačnega črkovanja v predobdelavi. Dobro bi bilo vključiti tudi emotikone in kontekst, v katerem se komentar pojavi. Poskrbeti je potrebno za hitrost, princip pa bi lahko uporabili tudi za druge jezike, na primer slovenščino.

# Chapter 1

# Introduction

There is a lot of hate speech online, as users hide behind anonymity and do not need to comply to social norms or think about the consequences of their words. We can face this problem on all online platforms that allow users' opinion to be shared to the general public. Companies hire people (called moderators) specifically for the purpose of deleting hateful comments on their website. They do this with the help of automated tools for detecting hate speech. These tools flag comments with possible hateful speech, and notify the moderators, which then make the final decision of deleting it or not. Such censorship, however, is contrary to the freedom of speech (the psychological aspect of this problem is discussed in more detail in [13]).

Instead of irreversibly removing user posts, we could detect hate speech right away and suggest an improved non-hostile version of the comment which preserves the original meaning. In this way we could avoid threats, insults, racism and discrimination in general. At the same time we would allow freedom of expression, provided that it is respectful of other participants in the conversation.

As verbal violence is not desirable, our goal is communication without profanity. As for the younger generations, a restriction of hate speech would create an environment for positive growth and development, while instilling the habits of good communication.

Such a solution would also greatly benefit companies and organizations that run online news sites, forums, social media platforms, and want to offer their users a nice, friendly environment to which they will like to return.

We developed two approaches for removing hate speech from online comments. The first approach uses paraphrasing to generate similar sentences and checks them for hate speech with a hate speech detector, while the second uses a T5 language model, which was fine-tuned for this task.

Chapter 2 starts by describing background and methods essential for the work and examines the previous work in the area of hate speech removal. Chapter 3 is split into two approaches developed for this problem, and presents details of the implementations. Chapter 4 describes evaluation of the proposed method, summarizes the results and analyses the cases when the algorithm fails. In chapter 5, we summarize the work and propose future improvements.

# Chapter 2

# Background

In this chapter, we present related work regarding the methodology used in this thesis. In Section 2.1, we present two social constructs hate speech and freedom of speech, laws regarding them and how they affected the online communities on different social media sites. Section 2.2 contains the overview of methods and technologies widely used in the field of natural language processing. In Section 2.3 we conclude the chapter with a short overview of work related specifically to the subject of removing hate speech in online comments.

## 2.1 Hate speech

According to the Cambridge Dictionary, hate speech is defined as "public speech that expresses hate or encourages violence toward a person or group based on something such as race, religion, sex, or sexual orientation" [14]. Usually it is considered hateful or hurtful communication towards a certain group with similar background or believes. Despite that, legal definition of hate speech is different in each country.

The law in certain countries describes hate speech as speech, conduct, gestures, writing, or displays that provoke violence or harm upon a certain group or individual members of a group; or the act of humiliation or intim-

idation of such people. The law can define these groups based on different characteristics that its members possess. In some countries the victims can press charges based on civil law, criminal law or even both. In other countries, the laws on hate speech do not exist. For example in the United States ideas (even those with intent to harm) are protected by the constitution and are considered freedom of speech.

### 2.1.1 Freedom of speech

Freedom of speech is a human right that supports the freedom of an individual or community to express their opinions and ideas without the fear of censorship or legal consequences. In legal terms, the phrase "freedom of speech" refers to seeking, obtaining and transmitting information or ideas, regardless of how or where they are used.

As widely recognized human right, freedom of speech is defined in the International human rights law (IHRL) and in article 19 of Universal declaration of human rights [15]. The latter expresses that "everyone shall have the right to hold opinions without interference" and "everyone shall have the right to freedom of expression; this right shall include freedom to seek, receive, and impart information and ideas of all kinds, regardless of frontiers, either orally, in writing or in print, in the form of art, or through any other media of his choice." The article was later improved in International Covenant on Civil and Political Rights (ICCPR) [16] by stating that exercising this right holds certain "special duties and responsibilities" and may "therefore be subject to certain restrictions" when necessary "[f]or respect of the rights or reputation of others" or "[f]or the protection of national security or of public order (order public), or of public health or morals."

### 2.1.2 Laws in Slovenia

In Slovenia, hate speech is not a legal term, even though the law describes such crimes. Article 297 of the Criminal Code of the Republic of Slovenia [6]

(KZ-1, 2008) defines several types of criminal offenses:

- public incitement to hatred, discord or intolerance based on ethnicity, racial, religious or ethnic origin, sex, color, origin, wealth, education, social status, political or other beliefs, disability, sexual orientation or any other personal circumstances
- publicly sharing the idea of superiority of one race over another or providing any assistance in racist activities
- denying, downplaying, approving, belittling, ridiculing or advocating genocide, the Holocaust, crimes against humanity, war crimes, aggression or other crimes against humanity.

In above cases the offender faces up to two years in prison. If they are co-responsible for the publication, the editors of the media are also subject to this threat. An aggravating circumstance is that if the acts are committed by an official, he or she faces up to five years in prison.

What makes the offence more serious is if an act described above is committed by coercion, ill-treatment, endangering security, disgracing national or religious symbols, damaging foreign objects, desecrating monuments, memorials or graves. In such case a sentence of up to three years in prison can be given.

### 2.1.3 Social media

To protect its users, social media sites incorporate rules regarding hate speech. In 2013, under pressure from more than 100 advocacy groups, including the Everyday Sexism Project, Facebook revised its hate speech policy after the release of data on content that encourages domestic and sexual violence against women. Consequently ads from 15 large businesses were withdrawn [17].

Later on May 31, 2016, Facebook, Google, Microsoft and Twitter agreed on European Union code of conduct by which they are required to review

"majority of valid notifications for removal of illegal hate speech", all within 24 hours [2].

Some of the companies that have adopted the hate speech policy are Facebook and YouTube. In 2018, a post containing parts of the US Declaration of Independence which referred to Indians as "merciless Indian savages" was flagged as hate speech on Facebook and removed from the website [18]. In 2019, the video platform YouTube demonetized channels like US radio host Jesse Lee Peterson, which supposedly violated their hate speech policy [19].

## 2.2 Natural language processing

The act of processing and analysing of textual data done by a machine is called natural language processing (NLP). The field of NLP encompasses many low level tasks, such as word segmentation, lemmatization, stemming, part of speech tagging and named entity recognition, and high level tasks like sentiment analysis, grammatical error correction, machine translation, natural language generation, natural language understanding, question answering, etc.

### 2.2.1 Rephrasal problem - paraphrase generation

Paraphrase generation is a task of generating sentences that are similar to the original text. Such tasks in automatic summarization and machine translation are usually evaluated with BLEU [20] or ROUGE [21] metrics. Because one can express the same idea with many different sequence of words, these measures give low scores to correct paraphrases due to a lexical mismatch. If we want to measure semantic similarity, a data-driven approach using a model that learnt semantic correlation between words in English language might be better.

The system in [22] produces similarities with the GAN (Generative Adversarial Networks) model, which is often used to generate photo-realistic images. The system uses one neural network to generate and another to

evaluate generated cases and returns a so called "reward". In such a feedback loop, both networks learn at the same time, which has lead to good results. Because of the rewards, the principle is similar to reinforcement learning. The GAN evaluator initially sees humanly annotated examples to know what positive examples look like. However, because there is no optimal way to express oneself in natural language, accurate assessment is more difficult. Therefore, the idea is enhanced with an additional neural network for learning the evaluator.

The best performing models for paraphrase generation for the last three years have all used the transformer architecture (link).

### 2.2.2 Hate speech detection

Hate speech detection is used on different platforms, from social media, news sites, to different forums. If one wants to create an online community, rules are inevitable and automation, especially on bigger websites, helps enforce them. Human moderators cannot keep up with the rate the comments are generated, thus algorithms label them, and hate labeled comments are forwarded to human evaluators to make the final decision, to either ignore them (in case of false positives) or remove them from the platform (in case of a true positives).

The solutions proposed to this problem aligns with the evolution of main NLP paradigms. First, text was classified by extracting features and then using one of the standard classifiers (logistic regression, naive Bayes, decision trees, random forests, SVM), with SVM giving promising results. When deep learning approaches were developed, the task was solved by learning word embeddings and doing classification on them. Deep neural networks used were CNNs which are good at feature extraction, RNNs, BiRNNs and LSTMs due to the sequential nature of the problem. And at the time of writing this thesis transformers dominate the field.

Most of the methods used before the transformers are compared in [3]. Authors evaluated machine learning methods like logistic regression, SVM,

gradient boosted decision trees (GBDT), neural networks like CNN and LSTM, and their combinations on Twitter comments. The best result was given by a combination of LSTM, random embbedings and GBDT, which achieved the F1-score of 93%. Still these methods cannot compare to transformer based techniques. Methods based on transformers (like BERT) achieve up to 96% F1-score [23] in the task of hate speech detection.

### 2.2.3 Transformers

The transformer neural networks were first introduced in 2017 [4] by the Google research team, gaining its true popularity in 2019.

Transformers are made from a encoder and decoder part. The encoder takes word embeddings as an input (which are independent of each other) and outputs the contextual embeddings, meaning that the vectors are changed so that they contain the information on the position in the sentence and how do the words correlate to each other. The decoder is built similarly to the encoder part, with the difference that it takes encodings (encoder outputs) into consideration and generates one word at a time from the previously generated ones. This process uses the attention mechanism in each layer. Both parts contain also a feed-forward neural network for additional processing and layer normalization steps.

### 2.2.4 Language models for transfer learning

Current transformer language models are used in transfer learning, where they use the knowledge gained from a previous task to improve generalization about another. One of the first transformer-based architectures to exploit that was BERT (Bi-directional Encoder Representations from Transformers) [5]. BERT uses a multi-layer bidirectional transformer encoder. Its self-attention layer performs self-attention in both directions of the input text. There are two variants of the model: BERT Base (with 12 layers and 110 million parameters) and BERT Large (with 24 layers and 340 million

parameters).

OpenAI team also took advantage of transfer learning to make their GPT language model [24]. Learning was separated into two parts, in the first step the model is learned unsupervised on random text, and in the second step we train this general model for a specific task (fine-tuning). The idea is that the model first learns the general language rules on unlabeled text, as there is a lot of it, and then uses valuable labeled data to learn specific tasks such as classification, assessing similarities, question answering, etc. As the basis, a multi-layer architecture consists of transformer decoders. GPT has been improved with larger and larger corpora of text in the first stage of training. GPT2 uses 1.5 billion parameters and GPT3 has 175 billion parameters. Both were initially withheld from the public because of alleged abuse threat.

A similar approach was developed by the Google AI team, in the text-to-text transfer transformer, T5 for short [25]. The difference is that T5 uses the whole transformer architecture (the encoder as well as the decoder part). Additionally, to make the model useful across tasks, they need to be transformed in to text-to-text problems (for example text classification would have labels positive/negative instead of a number 0/1). Multiple versions of the pre-trained T5 model were released, with the smallest having 60 million parameters and the biggest having 11 billion.

## 2.3 Related work

Removing hate from text has been researched by Salminen et al. [7]. The proposed concept focuses on deleting hateful elements in the text. In the first step, a hate speech detector is used. Based on the activations of this model, one determines which tokens (words) are hateful. To keep the sentence meaningful, not only the hate word is removed, but also all those that depend on it (dependence is evident from the dependency tree). Methods used are already a bit outdated. Because the method removes parts of the text, the authors suggest that a generative approach should be used in the future.

A similar topic dealing with the prevention of hate speech is called counter-speech / counter-narrative generation. Although this methodology also seeks to achieve a balance between hate speech and freedom of speech, it does not solve the problem in the same way. It is best described in a quote from [26]: "A Counter Narrative (CN) is a non-negative response to a Hate Speech (HS), targeting and contradicting extreme statements with fact-bound arguments or alternative viewpoints. Such a strategy seeks to de-escalate the conversation, disengage from hateful sentiment and encourage mutual understanding through an exchange of opinions.".

Whereas the first approach only deletes the hateful parts of the sentence thus limiting freedom of speech, we focus on generating a new sentence with the hateful part changed. The second approach just informs the user why they are wrong and not how to fix it.

# Chapter 3

# Hate speech removal

To create a system that will be able to convert hateful comments into normal ones, two approaches were used:

- On the collection of hateful comments, we can use a paraphraser to generate equivalent sentences, for which we check with a hate speech detector whether they are suitable,

- Fine-tuning of the T5 pre-trained language model is done on a set of hate-nonhate sentence pairs. Because such dataset do not exist, the dataset created in the previous approach was used.

Four main components are needed for the proposed system: a dataset of labeled hate speech comments, a pre-trained paraphraser, a pre-trained hate speech detector, and a pre-trained T5 language model.

A large number of pre-trained models and databases are available on the Hugging Face website, which serves as a repository for the NLP community. Anyone can publish their model on the platform, but the models (transformers) need to be written in Python. The end users can then download and use them in code or use the Hugging face API to integrate them into their service.

## 3.1 Component-based algorithm

By utilizing component-based development (a reuse-oriented approach in which independent components are combined into a complete system) techniques, we save time by not building the components from scratch (including training and testing) but using pre-trained state-of-the-art models.

### 3.1.1 Initial hate annotated dataset

The dataset of labeled hate speech Hatexplain [8] was already partially cleaned. Tokens like <user>, <percent>, <number> were used to eliminate unnecessary information. Because the hate speech detector cannot handle these tokens, we replaced them with random values ("percent" with decimal numbers from 0 to 100, "number" with integers from 1 to 2025 and "user" with a random English name). In the pre-processing part, we also removed emojis.

Each phrase in this dataset is labeled by three annotators as "normal", "offensive" or "hate". Because we need hateful inputs, only those with a majority vote "hate" were used.

These sentences are used as input to the algorithm, which shall return their non-hateful versions.

### 3.1.2 Paraphraser

For paraphrase generation, we used the Pegasus library [9]. By using different parameters, the paraphraser yields different results. We adjusted parameters like number of returned sequences (*num_return_sequences*), number of beams (*num_beams*), number of beam groups (*num_beam_groups*) and diversity penalty (*diversity_penalty*) to control how much the output sentences differ from the original.

We came to a (logical) conclusion confirmed in testing that more hateful the sentence is the more it has to be changed (higher diversity).

### 3.1.3 Hate speech detector - toxicity evaluator

To detect if a sentence is hateful, we used hate speech detector Detoxify [10]. This is a model called ALBERT trained on comments from Wikipedia (labeled 'original' in the documentation).

The model returns toxicity probability and the probability for different types of toxicity (toxicity, severe_toxicity, obscene, threat, insult, identity_hate). **The toxicity score does not indicate how toxic a certain sentence is, but only the probability if a sentence is toxic or not.** Therefore the model could not be used as a hate measure. In our case, sentences with the toxicity score 0.5 or above are labeled as hateful.

### 3.1.4 Similarity evaluator

Similarity was measured by using BERT for generating word embeddings and then calculating cosine similarity between representations of two sentences [11].

We used the BERT model from the Huggingface library to transform words to an internal representation, called dense vectors. These numerical representations of tokens are also called contextual word embeddings. Because each word is represented by at least one token, we get quite a few of them for a single sentence. By merging them all together, we create a semantic representation of the input sequence. Using a similarity metric like cosine similarity we can calculate the similarity between different sentences. Through thorough testing and human judgement a value of 0.57 was found to be a good threshold for this problem. Sentences with similarity score above 0.57 are labeled as similar.

As an example of how the approach correctly identifies similarity based on the meaning (semantics) instead of syntax we use the sentence “the king was happy”.

| "the king was happy" | similarity |
|---|---|
| "the ring was happy" | 0.8656 |
| "the emperor was happy" | 0.9518 |

### 3.1.5 DPhate system

Our system performs up to two paraphrasing. The second happens only if previous steps did not yield successful results (either all generated sentences are toxic, or none of the non-toxic ones is similar enough to the original text). In each paraphrasing, the results are checked for toxicity (if the generated sentence is hateful) and similarity (with the original text).

The pseudo code in Algorithm 1 takes as input a hateful sentence (string type) and returns a list of friendly sentences. In case it cannot generate any acceptable sentences, it returns an empty list. If a non hateful comment is passed as input, the algorithm just does a paraphrasing.

The input text is firstly decontracted (see the explanation below) and gets its vulgar adjectives and adverbers removed in lines 2 and 3. In lines 4 and 5, we then paraphrase the text and get the toxicity score of each newly generated sentence. In line 7, we compare the similarity of the original text and the generated sentences that are not toxic. Those that are not toxic and still similar are post-processed and returned in line 9. If the algorithm did not come to the return statement (either because every generated phrase was toxic or not similar enough), we take the least toxic of the paraphrased sentences and repeated the process between lines 4 to 11.

The Algorithm 1 uses the following methods:

- *decontract* - decontracts shortened versions of words such as “don’t”, “isn’t”, “won’t” and returns “do not”, “is not”, “will not”,
- *delete_vulgar* - removes vulgar adjectives and adverbs, which were found to carry little information. Vulgar words are identified by the *better_profanity* library, while the POS-tag is obtained by the *pos_tag*

method from the *NLTK* library,

- *paraphrasing* - returns a list of similar sentences, the parameters of the paraphraser change depending on the *toxCategory*, which is toxicity score grouped into four groups. The more toxic the input sentence, the more diverse the answers and larger the list that is returned as a consequence,
- *toxicity* - returns the toxicity score for each sentence in the input list,
- *similarity* - returns a list of similarity scores between the original phrase and the list of paraphrased non toxic comments,
- *post_processing* - removes all upper case sentences or those that contain a repeating pattern of an American phone number or the word "NationMaster". These are artifacts of the paraphraser which appear in some cases.

**Algorithm 1** Double Paraphrasing of hate speech

**function** DPHATE($text$)
    $textDecon \leftarrow decontract(text)$
    $newText \leftarrow delete\_vulgar(textDecon)$
    $paraList \leftarrow paraphrasing(newText, toxCategory)$
    $toxList \leftarrow toxicity(paraList)$
    **if** $any(toxList < 0.5)$ **then**
        $simList \leftarrow similarity(text, nonToxList)$
        **if** $any(simList > 0.57)$ **then**
            **return** $post\_processing(simNonToxList)$
        **end if**
    **end if**
    ▷ Second Paraprasing
    $minTox \leftarrow paraList[argmin(toxList)]$
    $paraList \leftarrow paraphrasing(minTox, toxCategory)$
    $toxList \leftarrow toxicity(paraList)$
    **if** $any(toxList < 0.5)$ **then**
        $simList \leftarrow similarity(text, nonToxList)$
        **if** $any(simList > 0.57)$ **then**
            **return** $post\_processing(simNonToxList)$
        **end if**
    **end if**
    **return** [ ]
**end function**

## 3.2 Paraphrasing with T5 model

We tested also the approach with T5 model, but it did not yield the same quality of results as the one described in the previous chapter. This is because the input for training were sentences generated by the previous approach, which were, at the time, not yet evaluated with a metric nor checked by human evaluators.

We used library Simple T5 [12] where the fine-tuning can be done in three lines of Python code (Figure 3.1). The figure shows the code for importing the library, loading the base model (a pre-trained general T5 model) and fine-tuning on a GPU with specific parameters. The final line of code is used for prediction, or rather generation of a non hateful version of the input sentence (it generates only one new sentence).

```
from simplet5 import SimpleT5

model = SimpleT5()
model.from_pretrained("t5","t5-base")

model.train(train_df=df[:index+1],
            eval_df=df[index+1:],
            source_max_token_len = 128,
            target_max_token_len = 128,
            batch_size = 3,
            max_epochs = 8,
            use_gpu = True,
            outputdir = "outputs")

model.predict(sentence)
```

Figure 3.1: The Python code for fine-tuning T5 model.

# Chapter 4

# Testing and results

In Section 4.1, we describe the automatic and human evaluation processes. The results of these evaluations are presented in Section 4.2. Problematic cases in which our proposed system fails and the limits of said system are described in Section 4.3.

## 4.1 Testing

For measuring quality of the generated text, two aspects were analysed, hateful content and similarity. Our goal is that the generated sentences are less hateful (offensive, preferably normal), while still similar to the original sentence.

We used similarity and toxicity score to measure quality of transformation.

When evaluating hate, we used an ensemble method called stacking. As baseline models, we used three pre-trained models [8, 27, 28] for detecting hate speech. Their predictions are passed to the meta learner, which is a multinomial logistic regression model. This method achieved an accuracy of 79.78% on two datasets [8, 29] combined, where accuracy is defined as the number of correct predictions divided by the total number of examples. Detailed measurements are shown in Figure 4.1. The confusion matrix is shown

in Figure 4.2, which shows the ratio of misclassifications for each class. Correct predictions are shown on the diagonal.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.79 | 0.68 | 0.73 | 2133 |
| 1 | 0.83 | 0.88 | 0.85 | 4754 |
| 2 | 0.68 | 0.66 | 0.67 | 1146 |
| accuracy | | | 0.80 | 8033 |
| macro avg | 0.77 | 0.74 | 0.75 | 8033 |
| weighted avg | 0.80 | 0.80 | 0.80 | 8033 |

Accuracy Score: 0.798

Figure 4.1: Different measures of the stacking algorithm for evaluating hate (0, 1 and 2 are indices for labeling Normal, Offensive and Hate speech respectively).

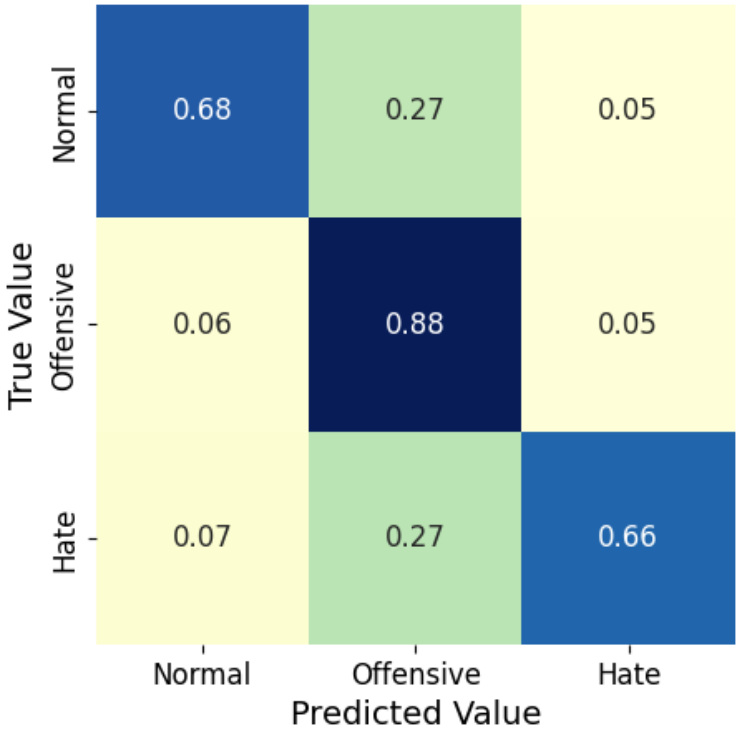


Figure 4.2: Confusion matrix of the stacking algorithm for evaluating hate.

Sentence similarity was evaluated using a pre-trained model called sim-CSE [30], created at the Princeton University. We used the supervised RoBERTa large model, which has an average accuracy score of 83.76% in the semantic textual similarity task and is at the time of writing the best model on the SICK dataset and the STS14 dataset (link).

### 4.1.1 Human evaluation

Because the models that give the automatic scores are not perfect, we also use a human evaluation of our final results.

To evaluate our approach, we published a campaign on the crowdsourcing platform microworkers.com where 61 people from English speaking countries applied and evaluated 876 examples of hate-nonhate sentence pairs, each

was labeled by 3 different evaluators. Their job was to label each pair on similarity on a scale from 1 (completely different meaning) to 5 (equivalent meaning) and evaluate hatefulness of the generated sentence also on a scale from 1 (friendly/neutral) to 5 (really hateful). The survey format is shown on picture 4.3.

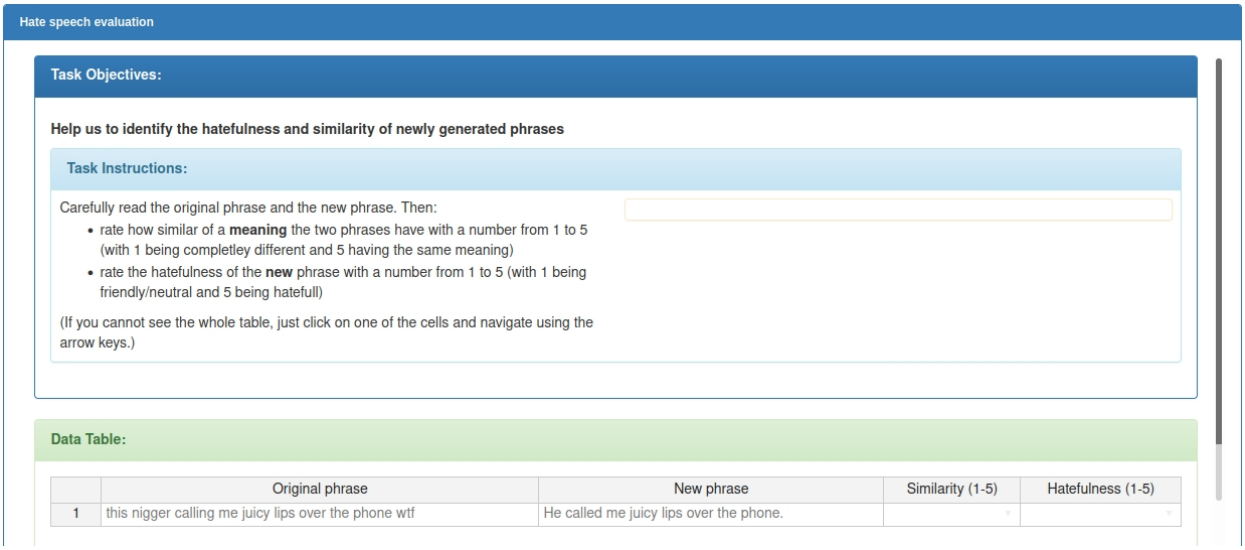


Figure 4.3: Picture of the survey as presented to the workers hired on the crowdsourcing platform MicroWorkers.com.

## 4.2 Results

The model for evaluating hate speech labeled 69% of generated sentences as normal, 13% as offensive and 18% as hateful (Figure 4.4). More importantly in 77.92% of sentence pairs the hatefulness has dropped (meaning the original sentence was labeled as hateful and the newly generated as offensive/normal, or the original labeled as offensive and the generated one as normal).

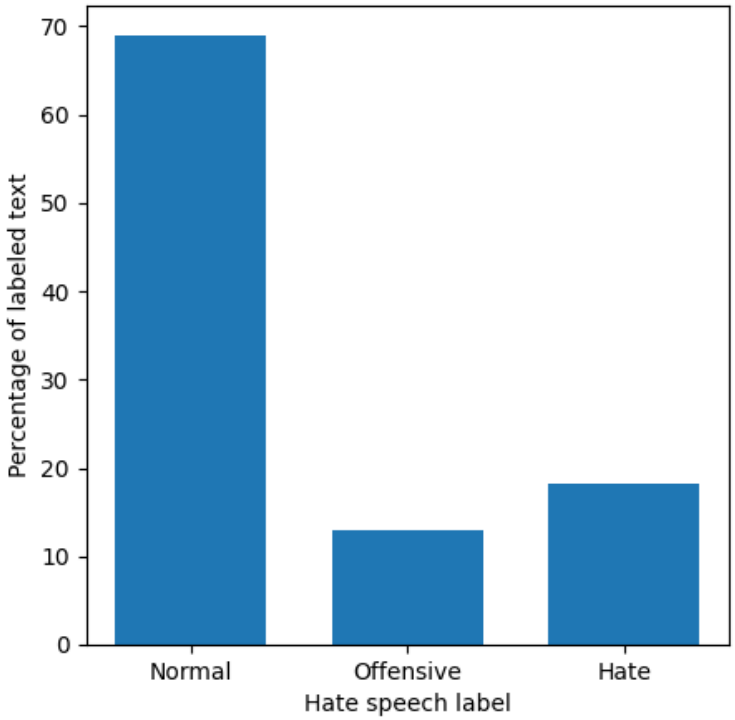


Figure 4.4: Bar chart showing the percentage of generated text labeled as hateful, offensive and normal.

The simCSE model labeled 92% of original-generated hate-nonhate pairs as similar and 8% as not similar, which is shown in Figure 4.5.

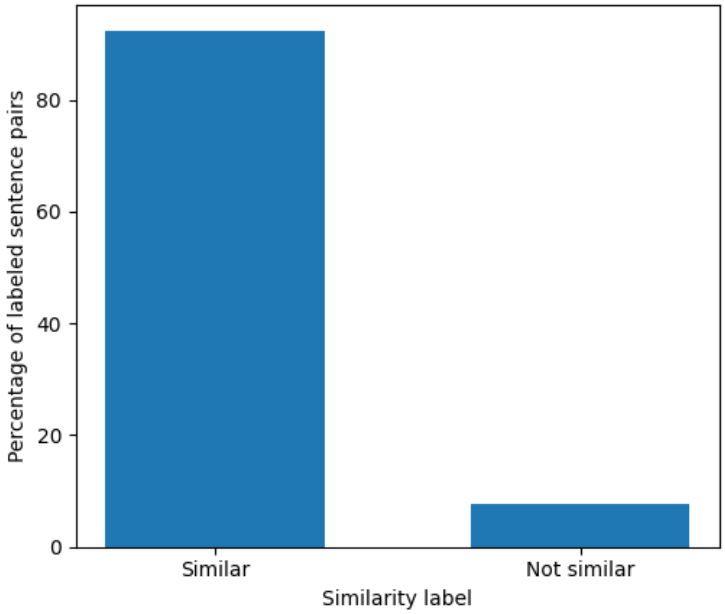


Figure 4.5: Bar chart showing the percentage of sentence pairs labeled as similar and dissimilar.

In conclusion, 70.74% of sentences got better (lower degree of hate and similar). **But because multiple "non-hateful" sentences were generated for each hateful input sentence (on average 12.64) the probability that at least one of those has lower hatefulness than the original while still being similar in meaning is 84.37%**.

Human evaluation through crowdsourcing concluded with less promising results as 81.77% of generated were labeled as non-hateful (on average they got a score of 3 or below) and only 51.26% as similar to the original text (on average they got a score of 3 or above); those that satisfy both conditions were 36.35%. **But again, due to generation of multiple viable solutions, the probability that at least one satisfies these conditions is 67.90%.**

## 4.3 Experimental observations

When reviewing the generated sentences we've identified situations in which the algorithm does not produce good results.

The majority of problems appear when the input contains slang, a misspelling or grammar error. Some examples are shown in the table below.

| **slang hate word** | **meaning** | **example input sentence** | **example output sentence** |
|---|---|---|---|
| dyke | lesbian (derogatory) | "only when middle school girls drive dykes to the nuthouse underage shitskin rapists get their ids protected" | "It's only when middle school girls drive dykes to the nuthouse." |
| dike | (same as dyke) | "being a disgusting fat dike should be illegal" | "It should be illegal to be a disgusting dike." |
| kike | jewish (derogatory) | "the only kike here is you" | "There is only one kike here, you." |
| muzzie | muslim (derogatory) | "says the swamp muzzie who fucks goats" | "says the swamp muzzie who does goat tricks." |
| mudshark | white woman who sleeps with black men (racial slur) | "also that he a faggot and serial mudshark" | "He is also a serial mudshark." |
| sandnigger | an Arab (derogatory) | "i can not tell one sand nigger from the next" | "I can't tell you from the next sandnigger." |
| wigger | white nigger | "Vera Ethan what a butthurt wigger" | "The wigger is Vera Ethan." |
| velcro head | a black person (derogatory) | "or velcro headed shit skin darkies " | "velcro headed darkies are what they are" |
| bich | bitch | "these nigger biches look like aunt jemima" | "The Biches looked similar to Aunt jemima." |

Sometimes the DPhate has a problem finding a good solution for short sentences, because after the hateful parts are removed they contain too little information to make a full sentence. Some examples are: "Ricardo is a faggot", "fuck this nigger bitch", "fuck that nasty bitch with a moslem dick", "Wade mentally retarded faggot". These comments are often just meant to offend others, and have no other context or background.

Hate speech detector can also tag normal sentences as hateful just because there are certain words that trigger it, even though they are used in a positive context. For example, the sentence "I like homosexuals" is flagged as hateful.

To generate new better sentences for 3,570 examples, DPhate took approximately 3 days on a quad core 3.1 GHz processor, or approx. 15 hours on a NVIDIA Tesla P100 GPU. The main bottleneck is the paraphraser, its time is mostly dependent on the input sentence's length, and the number of sentences it has to produce. The time doubles if the algorithm has to do the paraphrasing twice.

# Chapter 5

# Conclusion

In the mission to make the web a nicer place by helping people to express themself respectfully, we developed the DPhate system. The system makes sentences more pleasant to the recipient and acceptable to be posted on most social media sites. It initially deletes inherently vulgar words and generates multiple similar sentences (paraphrases) and checks how hateful and how similar are they to the original. If there is no good solution, it repeats the process on the least hateful paraphrase. The algorithm was tested automatically with an ensamble of pre-trained NLP models and by English speaking workers hired on the crowdsourcing platform MicroWorkers.com. In automatic evaluation, 84.37% of input sentences were found to have at least one generated sentence which sufficed the conditions (lower hate and similar meaning). Human evaluators considered 67.90% of sentences a good solution.

The DPhate system still has weak points.

In future work, we could convert slang words into their dictionary synonyms and fix misspelled words in the pre-processing stage. Emojies should be taken into account (they were excluded in the pre-processing part), as they often carry additional information and can hint to the sarcasm. People often respond to an article or to another post, thus context might help with better prediction and generation.

Another idea worth trying is training the models on data received from crowdsourcing, or to engage people to write friendly translations.

As people from all over the world use the web, the system could be made cross-lingual. The problem is the supporting architecture (like pre-trained models for paraphrasing, hate speech detection, etc.).

Real-time computing is important and unless we run multiple instances of the algorithm in parallel, it will not be fast enough to deploy it in any practical setting.

# Literature


[1] Gustavo S. Mesch. "Family Relations and the Internet: Exploring a Family Boundaries Approach". In: *Journal of Family Communication* 6.2 (2006), pp. 119–138. DOI: 10.1207/s15327698jfc0602_2.

[2] Alex Hern. "Facebook, YouTube, Twitter and Microsoft sign EU hate speech code". In: *The Guardian* (May 31, 2016). URL: https://www.theguardian.com/technology/2016/may/31/facebook-youtube-twitter-microsoft-eu-hate-speech-code (visited on 01/22/2022).

[3] Pinkesh Badjatiya et al. "Deep Learning for Hate Speech Detection in Tweets". In: *Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 2017, pp. 759–760. DOI: 10.1145/3041021.3054223.

[4] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017. ISBN: 9781510860964.

[5] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

[6] *Kazenski zakonik*. https://www.uradni-list.si/glasilo-uradni-list-rs/vsebina?urlid=200855&stevilka=2296. Accessed: 2021-06-29.

[7] Joni Salminen et al. "Neural Network Hate Deletion: Developing a Machine Learning Model to Eliminate Hate from Online Comments: 5th International Conference, INSCI 2018". In: Jan. 2018, pp. 25–39. DOI: 10.1007/978-3-030-01437-7_3.

[8] Binny Mathew et al. "HateXplain: A Benchmark Dataset for Explainable Hate Speech Detection". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.17 (May 2021), pp. 14867–14875.

[9] Jingqing Zhang et al. *PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization*. 2019. arXiv: 1912.08777 [cs.CL].

[10] Laura Hanu and Unitary team. *Detoxify*. Github. https://github.com/unitaryai/detoxify. 2020.

[11] James Briggs. *BERT for measuring text similarity*. 2021. URL: https://towardsdatascience.com/bert-for-measuring-text-similarity-eec91c6bf9e1.

[12] Shivanand Roy. *SimpleT5*. https://github.com/Shivanandroy/simpleT5. 2022.

[13] Stephen J. Ceci and Wendy M. Williams. "Who Decides What Is Acceptable Speech on Campus? Why Restricting Free Speech Is Not the Answer". In: *Perspectives on Psychological Science* 13.3 (2018). PMID: 29716456, pp. 299–323. DOI: 10.1177/1745691618767324.

[14] *hate speech*. URL: https://dictionary.cambridge.org/us/dictionary/english/hate-speech.

[15] U.N.G. Assembly. "Universal declaration of human rights". In: *Resolution adopted by the General Assembly* 10.12 (1948).

[16] United Nations (General Assembly). "International Covenant on Civil and Political Rights". In: *Treaty Series* 999 (Dec. 1966), p. 171.

[17] Sara C Nelson. "#FBrape: Will Facebook Heed Open Letter Protesting 'Endorsement Of Rape & Domestic Violence'?" In: *Huffington Post UK* (May 28, 2013). URL: https://www.huffingtonpost.co.uk/2013/05/28/fbrape-will-facebook-heed-open-letter-protesting-endorsement-rape-domestic-violence_n_3346520.html (visited on 01/20/2022).

[18] Sam Wolfson. "Facebook labels declaration of independence as 'hate speech'". In: *The Guardian* (July 5, 2018). URL: https://www.theguardian.com/world/2018/jul/05/facebook-declaration-of-independence-hate-speech (visited on 01/12/2022).

[19] Gregg Re. "YouTube ends monetization of conservative commentator Steven Crowder's channel, several others after left-wing outrage". In: *Fox News* (June 5, 2019). URL: https://www.foxnews.com/tech/youtube-steven-crowder-carlos-maza-vox-adpocalypse (visited on 01/03/2022).

[20] Kishore Papineni et al. "BLEU: a Method for Automatic Evaluation of Machine Translation". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2002, pp. 311–318. DOI: 10.3115/1073083.1073135.

[21] Chin-Yew Lin. "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Text Summarization Branches Out*. Association for Computational Linguistics, July 2004, pp. 74–81.

[22] Zichao Li et al. "Paraphrase Generation with Deep Reinforcement Learning". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018, pp. 3865–3878. DOI: 10.18653/v1/D18-1421.

[23] Hind S. Alatawi, Areej Alhothali, and Kawthar Moria. "Detection of Hate Speech using BERT and Hate Speech Word Embedding with Deep Model". In: *ArXiv* abs/2111.01515 (2021).

[24] Alec Radford et al. "Improving language understanding by generative pre-training". In: (2018). URL: https://openai.com/blog/language-unsupervised/.

[25] Colin Raffel et al. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *Journal of Machine Learning Research* 21.140 (2020), pp. 1–67.

[26] Yi-Ling Chung, Serra Sinem Tekiroğlu, and Marco Guerini. "Towards Knowledge-Grounded Counter Narrative Generation for Hate Speech". In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, 2021, pp. 899–914. DOI: 10.18653/v1/2021.findings-acl.79.

[27] Niklas von Boguszewski et al. "How Hateful are Movies? A Study and Prediction on Movie Subtitles". In: *Proceedings of the 17th Conference on Natural Language Processing*. KONVENS 2021 Organizers, pp. 37–48.

[28] Nakul Lakhotia. *Hate Speech Detection in Social Media in Python*. https://github.com/NakulLakhotia/Hate-Speech-Detection-in-Social-Media-using-Python. 2020.

[29] Thomas Davidson et al. "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. 2017, pp. 512–515.

[30] Tianyu Gao, Xingcheng Yao, and Danqi Chen. "SimCSE: Simple Contrastive Learning of Sentence Embeddings". In: *Empirical Methods in Natural Language Processing (EMNLP)*. 2021.