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

The web is full of hatred, 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. Usually because of the number of daily comments they use automated hate speech detection software to sift through it all. In this paper we propose a new system DPhate, which outputs a nicer alternative to the hateful comment, when a user tries posting it. The method uses a series of pre-trained models, that by paraphrasing generate non-hateful sentences. The technical evaluation has shown, that in 84.37% at least one acceptable sentence is generated, whereas only 67.90% were deemed acceptable by human evaluators.

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

Povzetek

Naslov: Parafraziranje sovražnega govora

Avtor: Drejc Pesjak

Splet je poln sovraštva, ki ga dodatno omogoča možnost ohranjati lastno anonimnost, in mnogi forumi kot tudi novičarske spletne strani se poskušajo braniti pred njim z gručo moderatorjev, 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 bi uporabniku, ob objavi sovražnega komentarja, predlagal prijaznejšo alternativo z ohranjenim pomenom. V metodi je uporabljenih več prednaučenih modelov, ki s parafraziranjem generirajo nesovražne povedi. Tehnična evalvacija pa je pokazala, da se v 84.37% generira vsaj en primeren stavek, medtem ko so jih človeški evalvatorji samo 67.90% ocenili za primerne.

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

Razširjeni povzetek

Vse mlajši 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 nerazvitosti, ne morajo obraniti in tako, kot majhne spužve srkajo vase vse vsebine, ki jim pridejo pod prste. Takšna izpostavljenost negativnim vplivom spreminja generacije v ljudstvo pesimističnih in sovražnih ljudi, katerih besednjak sestavljajo primarno slabe besede [1].

Spletne strani (forumi, razne družabne platforme in novičarske strani) vsak dan bijejo bitko proti sovražnemu govoru. Ker želijo ustvariti prijazno okolje za njihove uporabnike, morajo postaviti nekakšne meje oz. pravila komuniciranja. Takšna pravila so sedaj prisotna že na vsaki večji platformi, kot so na primer: Facebook, Youtube, Twitter, New York Times, Reddit ... 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 cel dan pregledujejo vsebino na njihovem spletnem mestu, in ukrepajo v primeru kršitve. Ker pa je teh komentarjev ponekod tudi več deset tisoč na dan, se podjetja dodatno poslužujejo avtomatiziranih sistemov za prepoznavo sovražnega govora v besedilu. Ti algoritmi sami po sebi niso 100 odstotno natančni, zato morajo označene komentarje še vseeno pregledati ljudje.

Za zaznavo sovražnega govora so v začetku bili uporabljeni splošni algo-

ritmi za strojno učenje, kot so odločitvena drevesa ali SVM, ki klasificirajo na podlagi značilk pridobljenih iz besedila. Kasneje je pridobivanje značilk že bilo zajeto v metodah, tu govorimo predvsem o nevronskih mrežah prilagojenih za obdelavo besedila, kot npr. CNN, RNN in LSTM. Čeprav so te metode producirale dobre rezultate (tudi z F1-oceno do 93%), pa niso primerljive z najsodobnejšimi "state-of-the-art" algoritmi. Dandanes na področju NLP (obdelava naravnega jezika) dominirajo arhitekture, ki temeljijo na tako imenovanih transformerjih. Transformer, enostavno rečeno, je sestavljen iz enkoderja in dekoderja, pri čemer prvi izlušči pomen, drugi pa lahko nato iz njega sestavi novo poved. En model, ki za osnovo uporablja vrsto enkoderjev transformerja, je BERT. BERT je najprej učen na veliki množici splošnega besedila in šele nato doučen na manjši množici za specifično nalogo; takšen način učenja se imenuje prenosno učenje (angl. transfer learning). Ta jezikovni model so razvili v podjetju Google, osnovna verzija BERT Base pa ima 12 slojev oz. 110 milijonov parametrov. Zaradi velikosti modela kot tudi množice pa je tak pristop zelo učinkovit v katerikolem problemu za obdelavo jezika, tudi za zaznavo sovraženga govora.

Težavo, ki jo predstavlja zaznava in odstranitev sovraštva pa je, da krši temeljno človeško pravico do izražanja. Gre za nekakšno bitko med sovražnim govorom in svobodo govora. Vsaka država si pravno formalno to drugače razlaga, medtem ko je v Sloveniji dovoljeno svobodno izražanje do mere, kjer ne razpihujemo sovraštva do neke skupine (kar je kaznivo dejanje), pa je v Združenih državah Amerike svoboda govora ustavna pravica, izraz sovražni govor pa formalno ne obstaja. Splet, ki ne pozna državnih meja, se bolj nagiba k ameriški različici, saj naj bi splet bil svobodno mesto, ki omogoča vsakemu posamezniku, ne glede na starost, spol ali raso, da izrazi svoje mnenje, četudi je v nasprotju z mišljenjem množice.

Namesto odstranjevanja sovražnih komentarjev, ki bi omejevalo pravico do svobodnega izražanja ali opustitev pravil, kar bi lahko bilo škodljivo za mentalno zdravje vseh prisotnih, v tem delu predlagamo boljšo rešitev. Ob oddaji komentarja na določeni platformi bi se v primeru, da je le-ta sovražen,

uporabniku priporočale prijazne različice njegove povedi, ki nosijo enak pomen. Podobno so raziskovali Joni Salminen in kolegi [2], kjer so se ukvarjali z odstranitvijo sovražnega dela povedi. To so dosegli z nevronsko mrežo za zaznavanje sovražnosti, iz katere se vidi, katere besede so aktivirale nevrone, da je prišla do takšne odločitve, nato pa so odstranili ves del stavka, ki je odvisen od teh besed (odvisnot so določili z jezikovnim drevesom odvisji). Pri takšnem odstranjevanju lahko pride do delne ali celotne izgube pomena povedi. Zato avtorji predlagajo, da bi bilo v nadalje potrebno razviti generativen pristop (kjer ne gre samo za odstranjevanje), na kar se osredotočamo tukaj.

V svojem delu smo se poslužili razvoja po delih, kjer s pomočjo ponovne uporabe združimo različne komponente v celoten sistem. Tako se z uporabo najsodobnejših pred-naučenih jezikovnih modelov izognemo odkrivanju tople vode in testiranju vsakega dela posebej. Kot osnovo smo vzeli podatkovno zbirko označenega sovražnega govora *Hatexplain*, 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 vse smeškote. Vsak izmed sovražnih primerov smo pognali čez 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 se identificira s pomočjo knjižnice *better_profanity*, besedna vrsta pa z metodo *pos_tag* iz knjižnice *nltk*). Iz te obdelane povedi se nato, s parafraziranjem, generira več podobnih povedi (za to smo uporabili knjižnico *Pegasus*, kjer lahko z nastavljanjem določenih parametrov, vplivamo na raznolikost generiranih povedi). Nove povedi se nato z detektorjem *Detoxify* preveri za sovražnost, nesovražne pa nato primerjamo z originalno povedjo za podobnost. Podobnost je izračunana s kosinusno podobnostjo na vektorskih vložitvah pridobljenih iz modela BERT. Na koncu še izločimo tiste, ki so bodisi napisane v celoti z velikimi črkami, vsebujejo besedo "NationMaster"

ali pa ameriško telefonsko številko; v vseh teh primerih gre za nekakšne artefakte modela, ki generira podobne povedi. V primeru, da metoda ni našla primerne rešitve (ker so vse generirane povedi še vedno sovražne ali pa niso več podobne originalni) se ves 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, ki pa ni vrnil tako obetajočih rezultatov, saj povedi na tej točki še niso bile ocenjene, niti s tehničnega vidika kot tudi ne s pomočjo človeških evalvatorjev. To sledi principu "garbage in garbage out", ki pravi, da je model lahko le tako dober, kot podatki, s katerimi ga učimo.

Evalvacija je potekala po dveh kriterijih, sovražnost in podobnost, sovražnost generirane povedi in podobnost le-te z originalno. Pri čemer je zaželena nizka vrednost prvega in visoka vrednost drugega kriterija. Evalvacije algoritma DPhate smo se lotili z dveh vidikov.

Pri tehnični evalvaciji smo uporabili princip, ki bazira na podatkih, tj. uporaba pred-naučenih jezikovnih modelov, ki so bili učeni na ogromnih podatkovnih zbirkah, kar omogoča na, da dosegajo boljše rezultate na problemih s področja NLP. Za evalvacijo sovražnosti smo izkoristili arhitekturo ansambelske metode "stacking", kjer so za osnovo vzeti trije pred-naučeni modeli za zaznavo sovražnega govora, njihove napovedi pa se nato posredujejo do meta modela (večrazredna logistična regresija), ki jih združi in vrne končni rezultat. Takšna arhitektura je dosegla napovedno natančnost 79.78%. Za evalvacijo podobnosti smo vzeli pred-naučen model simSCE, ki ima povprečno natančnost 83.76% in je v času pisanja najboljši model na podatkovnih zbirkah SICK in STS14. Tehnična evalvacija je pokazala, da se je 77.92% parom povedi zmanjšala sovražnost in da je 92% generiranih povedi podobnih originalnim. Dalje je pokazala, da je 70.74% povedi, ki zadostujejo obema kriterijema, oz. ker se za vsako vhodno poved generira več podobnih, je 84.37% možnosti, da bo vsaj ena zadoščala kriterijema.

Ker pa jezik ni formalno univerzalni zapis in ne obstaja optimalnega

načina za komuniciranje, je evalvacija na tem področju težavna. Zato se je dobro zanašati na instinkte in znanje ljudi, ki ta jezik uporabljajo že vso svojo življenje. Tako smo na crowdsourcing platformi MicroWorkers objavili kampanjo, na katero se je prijavilo 61 ljudi iz angleško govorečih držav in so ocenili 876 primerov, pri čemer so vsakega pregledali trije različni evalvatorji. V anketi so vsak od kriterijev lahko ocenili z oceno od 1 do 5. Človeški evalvatorji so 81.77% primerov, kot nesovražne in 51.26% generiranih povedi podobnim originalnim, zgolj 36.35% pa jih je zadoščalo obema kriterijema. Ker pa je vsakič generiranih več odgovorov, je možnost, da vsaj eden zadošča kriterijema, 67.90%.

Algoritem DPhate še zdaleč ni popoln. Večina problemov nastopi, če vhodno besedilo vsebuje sleng, ali pa napačno črkovane besede. Velikokrat ima težave najti rešitev 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 zgolj z namenom, da užalijo naslovnika in ne nosijo nobenega dodatnega pomena. Dodatnega problema pa ne predstavlja kvaliteta, ampak hitrost. Trenutni algoritem lahko za primer porabi tudi pol minute, kar je v današnjem svetu preveč. Glavno ozko grlo tukaj je model parafraziranje, čas generiranje pa je odvisen od dolžine vhodne povedi in koliko povedi mora generirati.

V prihodnje so mogoče še izboljšave, kot je pretvorba slengovskih besed v njihove slovarske sopomenke in popravljanje napačnega črkovanja v predobdelavi. Dobro bi bilo vključiti tudi smeškote in kontekst, v katerem se komentar pojavi. Poskrbeti je še potrebno za hitrost, princip pa bi lahko uporabili tudi za druge jezike, kot 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 virtually every online platform that allows the user's 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 of possible hateful origin, and notify the moderators, which then make the final decision of deleting it or not. Such censorship, however, is contrary to the fundamental human right of freedom of speech (the psychological aspect of the complexity of this problem is discussed in more detail in [3]).

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. This way we could avoid threats, insults, racism, discrimination in general; and at the same time leave the person free to express himself, provided that he is respectful of other participants in the conversation.

Any emotionally developed person would agree that verbal violence is not the solution, thus our goal is that adults would communicate without such profanity, constructively not destructively. As for the younger generation

online, such a restriction of hate speech would create an environment for positive growth and development, while instilling them with the habits of a good communication.

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

In our paper we developed two different approaches for removing hate from online comments. The first approach, by paraphrasing, generates similar sentences and checks them for hate 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 different research fields and methods that are essential for the work done in the next chapter, it also examines the previous work done in the task of hate removal. The third chapter is split into two approaches developed for this problem, in detail it states which technologies, libraries and tools were used for the both of the implementations. The fourth chapter than describes ways of evaluating the proposed method, it summarizes the results and analyses in which cases the the algorithm fails. The last chapter summarizes the work and proposes future improvements.

Chapter 2

Background

2.1 Psychology

2.1.1 Hate speech

According to 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" [4]. 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 it as speech, conduct, gestures, writing, or displays that provoke violence or harm upon a certain group or individuals members of a group; or the act of humiliation or intimidation of such people. The law can define these groups based on different characteristics that its members possess. In some countries the victims can even press charges based on civil law, criminal law or even both. Yet 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 all protected by the constitution and are considered as freedom of speech.

2.1.2 Freedom of speech

Freedom of speech is a fundamental 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 it is also defined in the International human rights law (IHRL) and in article 19 of Universal declaration of human rights [5]. 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) [6] 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."

2.1.3 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 [7] (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.

2.1.4 Social media

To protect its users leading social media sites needed to incorporate some 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 [8].

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 [9].

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 [10]. In 2019, the video platform YouTube demonetized channels like US radio host Jesse Lee Peterson, which supposedly violated their hate speech policy [11].

2.2 Computer science

2.2.1 Natural language processing

In the field of Computer Science sometimes we want to get data from text to analyse it. The act of processing and analysing of textual data done by a machine is called Natural language processing. The field of NLP encompasses many low level tasks, such as word segmentation, lemmatization, stemming, part of speech tagging, named entity recognition, and some high level tasks like sentiment analysis, automatic text memorization, grammatical error correction, machine translation, natural language generation, natural language understanding, question answering, ...

2.2.2 Rephrasal problem - paraphrase generation

Paraphrase generation is a task of generating sentences that are similar to the original text. Such tasks (automatic summarization and machine translation) are usually evaluated with BLEU or ROUGE metrics. But because you can express the same idea with a quite different sequence of words, it is likely that these measures give a low score to a correct paraphrase due to a lexical mismatch. If we want to measure semantic similarity a data-driven approach (a model that learnt semantic correlation between words in English language) might be better.

The system presented in the article "Paraphrase Generation with Deep Reinforcement Learning" [12] shares similarities with the GAN (Generative Adversarial Networks) model, which is used today to generate photo-realistic images. The system works by having one neural network to generate and another that evaluates generated cases and returns a so called "reward". In such a feedback loop, both networks learn at the same time, which has been proven to lead to good results. Because of this reward system, the principle is a type of 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.

2.2.3 Hate speech detection

Because the web is a free place people can write anything including profanity in the heat of an argument, thus methods for detecting hate speech have been developed. Hate speech detection is now widely used on different platforms, from social media, news sites, to different forums. If one wants to create a great online community, rules are inevitable and automation, especially on bigger websites, helps enforce them. Human moderators just cannot keep up with the rate the comments get generated, thus fast algorithms label each of them, and because they are not still not perfect, the hatefully labeled comments get forwarded to human evaluators to make the final decision, to either ignore it (in case of false negative) or remove it from the platform entirely (in case of a true negative).

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 classic classifiers (logistic regression, naive Bayes, decision trees, random forests, SVM), with SVM giving very promising results. Later when deep learning approaches were developed, the task was solved by learning word embeddings and doing classification on them all within the same algorithm. At first deep neural networks were used, such as CNNs which are good at feature extracting or RNNs, BiRNNs and LSTMs due to the sequential nature of the problem. And at the time of writing this paper transformers dominate the field.

Most of the methods before transformers are compared in [13]. In the paper they 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 a F1-score of 93%. Still these methods cannot compare to transformer based techniques in the NLP space, just because of the deeper understanding of the language the latter has. Methods based on it (like BERT) achieve up to 96% F1-score [14] in the task of hate speech detection.

2.2.4 Transformers

Transformers were first introduced in 2017 in the paper "Attention is all you need" [15] by the Google research team, gaining its true popularity two years later, in 2019.

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

2.2.5 Language models

Current language models use a technique, called 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) [16]. BERT uses a multi-layer bidirectional transformer encoder. Its self-attention layer performs self-attention in both directions. Google has released two variants of the model: BERT Base (with 12 layers, which is 110

million total parameters) and BERT Large (with 24 layers or 340 million parameters).

In [17] the OpenAI team also takes advantage of transfer learning to make their language model more effective. More specifically, learning is separated into two parts, in the first step the model is learned unsupervised on random text, and in the second step we learn this general model for a specific task (fine-tuning). The idea is that the model first learns the general laws of language in unlabeled text, as there is a lot of it, and then uses valuable labeled data to learn for tasks such as: classification, assessing similarities, question answering... As the basis a multi-layer architecture is used consisting of transformer decoders. This so called GPT, has since then been improved by giving it larger and larger corpuses of text in the first stage of training, out of which came GPT2 with 1.5 billion parameters and GPT3 with 175 billion parameters. Both of which were initially withheld from the public because of the danger they pose if abused.

A quite similar approach was developed by the google AI team, the so called text-to-text transfer transformer, T5 for short [18]. The difference is that it uses the whole transformer architecture (the encoder as well as the decoder part). Additionally to make the model interchangeable between 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 Joni Salminen et al. [2]. The proposed concept focuses on deleting hateful elements in the text. In the first step, a hate speech detector is used, and based on the activations of this model, we determine 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). The problem is that the methods used are already a bit outdated, as the article was published in 2018. Because this method only removes parts of the text, at the end of the paper 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 [19]: "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.".

Chapter 3

Hate removal

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

- From the collection of hateful comments we can use a paraphraser to generate equivalent sentences, for which we check with a hatespeech detector whether they are suitable for publication.

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

From the previous point, it is evident that four main components are needed for the 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 sort of a large repository for the NLP community. Anyone can publish their model on the platform, still these 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 in 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 can 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 [20] was already partially cleaned, because 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 for simplicity.

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

These sentences are used as input in the algorithm, which supposedly returns friendly versions.

3.1.2 Paraphraser

For paraphrase generation we used the Pegasus library [21]. With using different parameters the paraphraser yields different results. By adjusting 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*) we can control how much the output sentences differ from the original.

We came to a logical conclusion which was confirmed by our testing results that more hateful the sentence is the more you need to change it (higher diversity).

3.1.3 Hate speech detector - toxicity evaluator

To detect if a sentence is hateful, we used, hate speech detector Detoxify [22], more specifically it is a type of a NLP 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 it could not be used as an actual hate measure. In our case any sentence with a toxicity score 0.5 or above is labeled as hateful.

3.1.4 Similarity evaluator

Similarity was measured by using BERT for generating word embeddings and then calculating cosine similarity between these representations of two different sentences [23].

Essentially a BERT model from the Huggingface NLP library is used to transform words to an internal representation, called dense vectors. These numerical representation of a single token are also called contextual word embeddings. And because each word is represented by one token, we get quite a few of them for a single sentence. By transforming them all together we create a semantic representation of the input sequence. Then by using a similarity metric like cosine similarity we can calculate the similarity between different sentences.

As an example of how it correctly identifies similarity based on 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 Algorithm DPhate

Essentially the algorithm performs two paraphrasing. The second one happens only if previous steps did not yield 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 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. And if a non hateful comment is passed as input, the algorithm just does a simple paraphrasing.

The input text is firstly decontracted and gets its vulgar adjectives and adverbers removed in lines 2 and 3. In lines 4 and 5 we then paraphrase that text and get the toxicity score of each newly generated sentence. In line 7 we compare the similarity of the original text and those of the generated sentences that are not toxic. Those that are not toxic and still similar are than 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 of lines 4 to 11.

The algorithm 1 uses 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 through data analysis were found to carry the least information, vulgar words are identified by the *better_profanity* library, while the part of speech tag is given 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 are 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 hate

```
function DPhate(text)
    textDecon ← decontract(text)
    newText ← delete_vulgar(textDecon)
    paraList ← paraphrasing(newText, toxCategory)
    toxList ← toxicity(paraList)
    if any(toxList < 0.5) then
        simList ← similarity(text, nonToxList)
        if any(simList > 0.57) then
            return post_processing(simNonToxList)
        end if
    end if
                                              ▷ Second Paraprasing
    minTox ← paraList[argmin(toxList)]
    paraList ← paraphrasing(minTox, toxCategory)
    toxList ← toxicity(paraList)
    if any(toxList < 0.5) then
        simList ← similarity(text, nonToxList)
        if any(simList > 0.57) then
            return post_processing(simNonToxList)
        end if
    end if
    return [ ]
end function
```

3.2 Language model

Another approach was tested, but it did not yield quite the same quality 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. This aligns with the "garbage in garbage out" principle, which states that the result can only be as good as the input data provided to the algorithm.

The used library Simple T5 [24] is very easy to use, as the fine-tuning can be done in quick three lines of Python code (Figure 3.1). The figure shows the code for importing the library, then 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 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 whole Python code for fine-tuning T5 model in 3 lines.

Chapter 4

Testing and results

4.1 Testing

For measuring quality of the generated text, two aspects were analysed, hate and similarity. Like in the building of the DPhate algorithm, we resorted to a data driven approach in the evaluation stage. With models that were trained on large corpuses of text we are able to get a better score when dealing with NLP tasks.

4.1.1 Technical evaluation

In the algorithm we used similarity and toxicity score to measure quality of translation. Initially also a grammar checker was used, but later abandoned due to the paraphraser's good design, where grammar was never really a problem. Similarly a model for analysing hate and a separate for similarity were used in the technical evaluation.

When evaluating hate we used an ensemble method called stacking. In the base part there are three pre-trained models [20, 25, 26] for detecting hate speech, their predictions are then passed to the meta learner, which is a multinomial logistic regression model. This method achieved an accuracy of 79.78% on two datasets [20, 27] combined, where accuracy is defined as the number of correct predictions divided by the total number of examples.

More specific measurement is shown in Figure 4.1 and the confusion matrix in Figure 4.2, which shows the ratio of misclassifications for each class, an optimal prediction is when the diagonal (from top left to bottom right) contains all ones and every other square contains a zero.

	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)

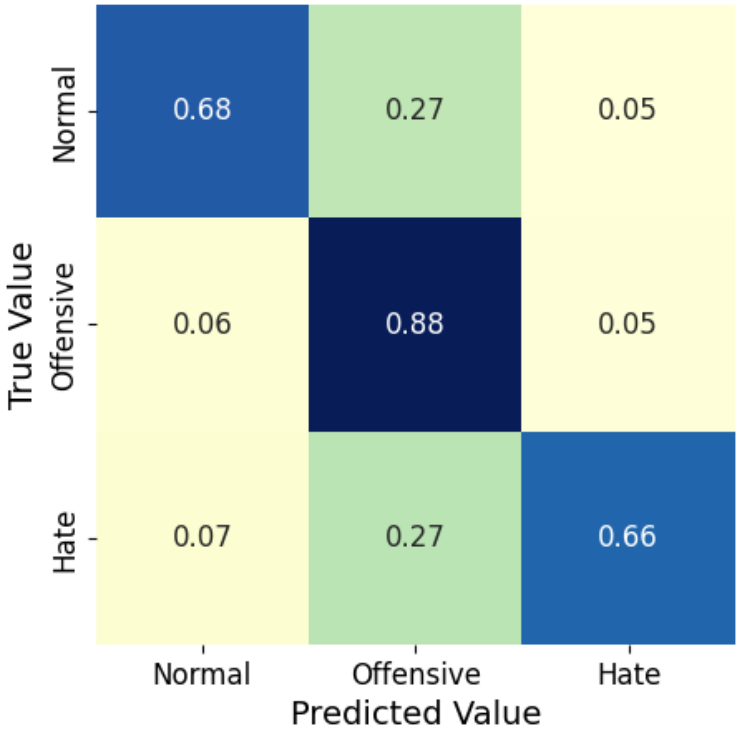

Figure 4.2: Confusion matrix of the stacking algorithm for evaluating hate

Similarity was evaluated using a pre-trained model called simCSE [28], 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 number one model on the SICK dataset and the STS14 dataset (link).

4.1.2 Human evaluation

As an intermediate solution this technical evaluation works, but because the models that give the scores are not perfect, it cannot be used as a formal metric.

Because natural language is not in itself a formally universal system and there are not any optimal ways to communicate, evaluating systems in this

field is difficult. It is best to rely on the instincts of people who have used these languages all their lives, so human evaluators were needed to assess the quality of the text conversion.

To properly evaluate the 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.

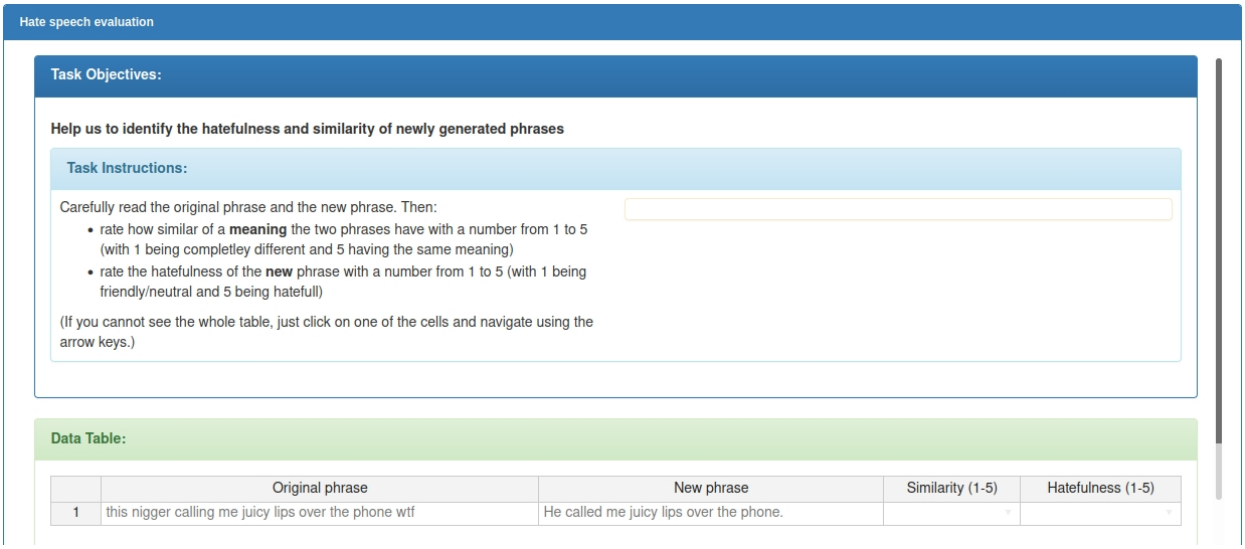

Figure 4.3: Picture of the survey as presented to the workers hired on the crowdsourcing platform MicroWorkers.com

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).

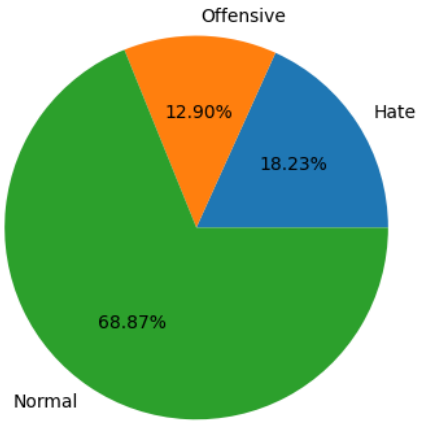

Figure 4.4: Pie graph showing the percentage of generated text labeled as hateful, offensive and normal.

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.

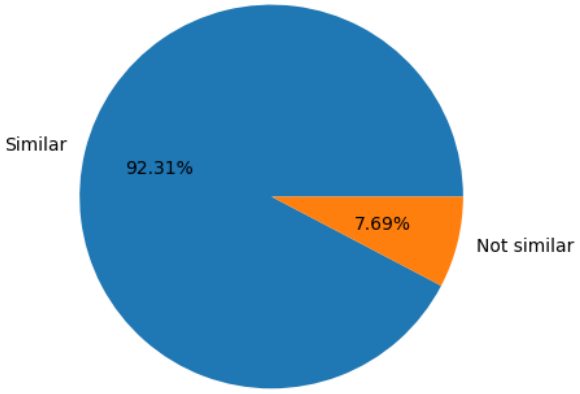

Figure 4.5: Pie graph showing the percentage of sentence pairs labeled as similar and dissimilar.

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 percentage 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 chance that at least one justifies these conditions is 67.90%.**

4.3 Experimental observations

When reviewing the generated sentences we've identified situations in which the algorithm still 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 A-rab (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 algorithm also has a problem finding a good solution for short sentences, this is 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.

To the contrary hate speech detector can tag friendly 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.

One thing that was not specifically mentioned was the speed of the DPhate algorithm. To generate new better sentences for 3,570 examples it 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, the time is mostly dependent on the input sentence’s length, and how many new sentences it has to produce. The time even 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 them self respectfully we developed the DPhate algorithm which can make these sentences more pleasant to the recipient and acceptable to be posted on most social media sites. It achieves this by initially deleting inherently vulgar words and then generating multiple similar sentences (paraphrase generation) and checking how hateful and how similar are they to the original. If there is no good solution, it repeats the process the second time on the least hateful answer. The algorithm was tested by a data-driven technical approach (a group of pre-trained NLP models) and also by english speaking workers hired on the crowdsourcing platform MicroWorkers.com. In technical evaluation 84.37% of input sentences were found to have at least one generated sentence which sufficed the conditions (lower hate and similar meaning). Whereas only 67.90% had a good solution according to the human evaluators.

5.1 Future work

As discussed in the previous chapter the DPhate algorithm still has some weak points.

Future work could include converting slang words into their dictionary synonyms and fixing misspelled words in the pre-processing stage. Also emo-

jies should probably be taken into account (which were excluded in the pre-processing part of our research), as they often carry additional information and can hint the sarcasm. A lot of the time people are not creating content from ground up, but are responding 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 even better get people to write friendly translations.

Furthermore, because people from all over the world use the web, the researched principle could be extrapolated to other languages. The problem is the small amount of supporting architecture (like pre-trained models for paraphrasing, hate speech detection, ...) that has been built till now.

Nowadays in commercial software real-time computing is very important and unless you run multiple instances of the algorithm in parallel it simply will not be fast enough for any serious company to deploy it as a feature on their website.

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. URL: https://doi.org/10.1207/s15327698jfc0602_2.

[2] Joni Salminen et al. "Neural Network Hate Deletion: Developing a Machine Learning Model to Eliminate Hate from Online Comments: 5th International Conference, INSCI 2018, St. Petersburg, Russia, October 24–26, 2018, Proceedings". In: Jan. 2018, pp. 25–39. ISBN: 978-3-030-01436-0. DOI: 10.1007/978-3-030-01437-7_3.

[3] 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. eprint: https://doi.org/10.1177/1745691618767324. URL: https://doi.org/10.1177/1745691618767324.

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

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

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

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

[8] 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).

[9] 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).

[10] 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).

[11] 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).

[12] Zichao Li et al. "Paraphrase generation with deep reinforcement learning". In: *arXiv preprint arXiv:1711.00279* (2017).

[13] Pinkesh Badjatiya et al. "Deep Learning for Hate Speech Detection in Tweets". In: *Proceedings of the 26th International Conference on World Wide Web Companion* (2017).

[14] 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).

[15] Ashish Vaswani et al. “Attention is All you Need”. In: *ArXiv* abs/1706.03762 (2017).

[16] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *ArXiv* abs/1810.04805 (2019).

[17] Alec Radford et al. “Improving language understanding by generative pre-training”. In: (2018).

[18] 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. URL: http://jmlr.org/papers/v21/20-074.html.

[19] Yi-Ling Chung, Serra Sinem Tekiroglu, and Marco Guerini. *Towards Knowledge-Grounded Counter Narrative Generation for Hate Speech.* 2021. arXiv: 2106.11783 [cs.CL].

[20] Binny Mathew et al. “HateXplain: A Benchmark Dataset for Explainable Hate Speech Detection”. In: *arXiv preprint arXiv:2012.10289* (2020).

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

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

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

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

[25] Niklas von Boguszewski et al. “How Hateful are Movies? A Study and Prediction on Movie Subtitles”. In: *arXiv preprint arXiv:2108.10724* (2021).

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

[27] 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. Montreal, Canada, 2017, pp. 512–515.

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