

CSE508: Information Retrieval
Lexical Text Simplification and Detoxification

Aanya Trehan 2020419	Abhimanyu Bhatnagar 2020273	Aniket Goel 2020281	Apoorva Arya 2020032	Naman Kaushik 2020088	Sejal Kardam 2020467
-------------------------	--------------------------------	------------------------	-------------------------	--------------------------	-------------------------

1. Updated Problem Statement

Our group aims to build a system to replace complex and difficult to understand words in a text with less complex words. However, developing effective lexical simplification techniques remains a challenging task due to the complexity of language and the nuances of meaning that can be lost in the simplification process. Therefore, we aim to develop a novel technique that can accurately and effectively simplify the text while maintaining the meaning and preserving the intended tone and style of the original text. We intend to make it more inclusive by incorporating different languages like Hindi and if possible a few regional languages. Additionally, we would also be working towards detecting offensive text to detoxify it so that it can be viewed by a large audience without repercussions. This would require developing sophisticated algorithms and models that can accurately identify offensive text and thereby, try to rephrase the text either by substitution or generation to output a “detoxified” version of the text.

2. Motivation

Many individuals with different cognitive and linguistic abilities struggle to understand the complexity of texts, which limits their access to information and impedes their ability to participate in daily activities. Lexical Text Simplification aims to address this challenge by automatically simplifying the language and structure of text while preserving the original meaning and intent of the message. Moreover, in today's society, there is a growing concern over the increase in hate speech which introduces the need for the detoxification of the text. The detoxification process moderates the content available online and makes it less toxic and relatively safer to be out there on the web.

3. Literature Review

3.1 TextRazor

TextRazor is a natural language processing (NLP) platform that utilizes machine learning algorithms to extract information from unstructured text, including tweets, web pages, and other text-based databases. However, it has limitations, such as being trained only on general text, which means it may not be able to analyze data in specialized domains, like scientific or technical

texts. Additionally, it struggles to handle texts with complex sentence structures and has limited language support. Moreover, TextRazor is a paid service, which may not be feasible for users with limited budgets.

3.2 HateSonar

HateSonar is an NLP tool that is specifically designed to detect hate speech in text. It analyses text using ML techniques to find linguistic patterns frequently found in hate speech. It gives false positive results as the models are not appropriately trained [1]. It performs poorly on our random sample with a precision of 0.5 and recall of 0.31 [2]. It provides limited language support and limited customization and is unable to identify some forms of hate speech, such as bullying or threats.

3.3 Perspective API Google

Perspective is a technology developed by Google that utilizes machine learning algorithms to detect abusive comments by scoring the impact of a given text in a conversation. Developers and publishers can use this score to provide feedback to commenters, make it easier for moderators to review comments, or allow readers to filter out offensive language [3]. However, An adversary can subtly modify a highly toxic phrase so that the system assigns a significantly lower toxicity score to it [4]. It faces difficulty in handling sarcasm and irony and difficulty keeping up with rapidly evolving languages and slang used in online conversations.

3.4 Rewordify

Rewordify is a web tool that makes difficult English easier to understand. It replaces hard English words and phrases with simplified forms [5]. This tool is designed primarily to simplify low-frequency difficult words; hence when sentences are made using simple high-frequency words, Rewordify cannot simplify the sentence further [6]. It also struggles in paraphrasing while retaining the context in certain scenarios like academic work with specific scientific language [7].

3.5 Simplish

Simplish is a tool that simplifies by rewriting text using fixed 1000 words of basic English vocabulary. It also tries to explain the difficult words using these basic English words. What makes it unique is the fact that instead of

replacing only difficult words or phrases in a text, it replaces all the words [8]. This can lead to a few problems, such as over-simplification of the text and changing context. Another issue is when the algorithm is unable to replace a term, it substitutes it with its dictionary definition instead. This results in an extremely long output that might be more difficult to understand. Its paid model offers more features, but that is not accessible to everyone.

3.6 Detoxification

The available detoxifying techniques are exclusive to one particular language and are not easily adaptable to other languages. Research shows that multilingual models are capable of transferring styles between different languages. However, evaluations show that they were unable to perform detoxification across many languages [9]. Detoxification methods have been proven to be detrimental to equity since they reduce the usefulness of LMs for the language used by excluded groups [10]. Particularly when it comes to the language used by marginalized groups, detoxification renders language models more susceptible to distribution change [10].

3.7 Civil Rephrases Of Toxic Texts With Self-Supervised Transformers

The research paper[13] proposes a method for detecting and rephrasing toxic or harmful texts using self-supervised transformers. The method involves utilizing a pre-trained transformer model to identify and replace toxic words and phrases with neutral or positive alternatives while preserving the original meaning of the text. The proposed approach is context-aware, allowing for the replacement of phrases based on their context in the text. The authors also explore the potential for applying the model to languages other than English, such as Hindi and Spanish. The proposed method has the potential to improve the inclusiveness and respectfulness of text, particularly in online communication where toxic language can cause harm.

4. Proposed Method

4.1 Lexical Text Simplification

We divided the task of lexical simplification into two major steps: identifying the complex words in a sentence and then finding appropriate words to replace these complex words.

4.1.1 Identifying complex words

For this we have used the function ‘difficult_words’ from the textstat library. The textstat library has a list of simple

english words. The ‘difficult_words’ function returns a word as complex if it is not present in the list of simple english words and has 3 or more than 3 letters.

4.1.2 Finding replacement words

Sentences containing more than 6 words:

In this case we first masked the identified complex word and then utilized the MLM (Masked language modeling) provided by BERT (by HuggingFace) to predict the appropriate words in place of the masked word. The BERT model gives a list of appropriate words.

To narrow down the list, we first eliminated all the words which were more complex than the original word. We used the zipf frequency of the words to do this. Words having lower zipf frequency are more complex than words having higher zipf frequency. Thereafter the best 2 words from the list were chosen on the basis of similarity to the original word. For this the complex word in each sentence was replaced with one of the proposed new words, one by one, to obtain simplified sentences. The original and modified sentences were transformed using the Bert Similarity Model. Then the similarity between the original sentence and the modified sentence was found using cosine_similarity. The best two words were chosen on the basis of maximum cosine similarity between the original and the modified sentences.

In case of multiple complex sentences in a word, all the possible permutations of the best 2 word lists of all the complex words were considered. So if there are 5 complex words in a sentence then there can be $2^5(32)$ combinations of the possible replacement words. Modified sentences corresponding to each of the permutations are obtained.

The most appropriate set of replacement words is chosen by taking into account the complexity of the modified sentence as well as the complexity of words in the sentence. The complexity of the modified sentences are obtained by summing up the zipf frequency of all the words in the sentence. The original and modified sentences were transformed using the Bert Similarity Model. The similarity between the original and modified sentences were obtained using cosine_similarity. A rank(starting from 1) was given to each sentence based on its simplicity(higher the simplicity lower the rank) and based on similarity to original text(higher the similarity lower the rank). For each sentence the both types of ranks were multiplied. The set of words that created the sentence which had the minimum product of the ranks was finally chosen.

Sentences containing less than 6 words:

If a sentence contained less than 6 words, then the identified complex words were replaced with the most appropriate synonyms. The reason for this is it is difficult

for the BERT model to identify appropriate replacement words if sufficient context is not available.

For this synonyms of the complex word were found using the pre-trained GloVe model which was loaded from the Gensim library. This model generated a list of synonyms. The most appropriate synonym was chosen by the rank product method using the ranks generated with the help of zipf frequency and cosine similarity as done in the greater than 6 words method.

Additionally if a sentence has greater than six words and has an adjective followed by a noun then the method of synonym replacement is used for that particular word and the BERT model is used for the rest of the complex words.

4.2 Detoxification

For our ultimate model, we conducted extensive experimentation with language generation models and various large language models (LLMs). However, we encountered a hindrance as there was no LLM readily available for detoxification tasks. To develop a new LLM, a comprehensive dataset would have been necessary, but the absence of such a dataset posed a significant challenge due to the subjectivity inherent in toxicity. The primary drawback of conventional techniques lies in the likelihood of losing semantic meaning, which led us to bifurcate the problem into two tasks: the initial objective being the removal or replacement of toxic verbiage, and the secondary objective being the preservation of semantic meaning in the original text.

Drawing inspiration from the research paper titled "Text Detoxification using Large Pre-trained Neural Models," we incorporated the strategy of synonym replacement and paraphrasing.

4.2.1 Paraphraser

In this regard, we attempted to deploy the para-Gedi model that was previously employed for pre-training a neural network [14]. Furthermore, we utilized a pre-trained T5 model to construct our own paraphraser, working in tandem with the AutoModelForSeq2SeqLM model. Our paraphraser accepts a context vector (i.e., the original sentence for which the context must be maintained) and an input sentence (i.e., the sentence requiring paraphrasing) to generate possible paraphrased sentences. As an example, we executed our paraphraser on the original sentence itself.

4.2.2 Synonym Replacer

The input text is first analyzed to detect any offensive or toxic words using a model trained on the Jigsaw dataset.

Once the toxic words are identified, they are classified into one of six categories based on the type of toxicity. For each category, a mapping of potential synonyms is developed, which can be used to replace the toxic words. These synonyms are semantically similar but less toxic than the original offensive words.

After the mapping of synonyms is created, the toxic words are replaced with each of the mapped synonyms. The new sentence is chosen as input for the paraphrasing step. The paraphrasing model then rewrites the input sentence in simpler language while preserving the meaning and tone of the original text. The resulting detoxified text is then provided as output.

5. Evaluation

We calculate the sum of Zipf frequency of all the words in the original and modified sentence to compare their complexity. Zipf frequency of a word is the base-10 logarithm of the number of times it appears per billion words. So higher the sum of the Zipf frequency, the more frequently used words the sentence has, the simpler the sentence is.

To compare the similarity of the two sentences, both the sentences are transformed using the Bert Similarity Model. Thereafter the cosine similarity is calculated between the two sentences.

6. Novelty

The detoxification of toxic text is an area that has been severely neglected in research. In our project, we have implemented a novel technique that combines four distinct tasks to effectively remove toxic content from text. The first task involves detecting toxic words, which is followed by categorizing these words into specific labels. We then utilize a mapping technique to associate these toxic words with their most appropriate synonyms, achieving high accuracy in the process. Finally, we utilize a paraphrasing technique that ensures the original context of the text is retained or restored, depending on the input. While each of these tasks has been implemented before, our project is unique in that it combines them for text generation.

Our results have exceeded expectations, demonstrating the effectiveness of our approach to detoxifying toxic text.

7. Database and Code:

7.1 Lexical Text Simplification

Database: The 'bert-large-uncased' model is used which is trained on a dataset called the BooksCorpus dataset. This dataset contains words from various english books as well as the english version of wikipedia.

The "glove-wiki-gigaword-50" model used for finding synonyms is trained on dataset

The code can be found here : [Link](#)

7.2 Detoxification

We have used the Jigsaw Unintended Bias in Toxicity Classification Dataset for training purposes. We have also used a pre-trained T5 model (an LLM model) for the paraphrasing section in order to retain the context of the input text. The code first detoxifies the input sentence and then moves on to the paraphrasing in order to bring back any context which was lost due to synonym replacement during the detoxification.

The Dataset can be found here: [Link](#)

The code can be found here: [Link](#)

8. Results:

8.1 Lexical Text Simplification

Given here are some outputs corresponding the inputs we fed into our model for text simplification:

Original text is: Agree to my demands or I will intimidate you.

Simplified text is: agree to my conditions or I will hurt you.

Simplicity of original text is: 53.75

Simplicity of simplified text is : 55.95

Cosine similarity between the two sentences is 0.903

Original text is: The door to his dilapidated residence was locked but the police knocked down the door with full force and captured him.

Simplified text is: The door to his run-down home was locked but the police knocked down the door with full force and arrested him.

Simplicity of original text is: 123.61

Simplicity of simplified text is : 133.58

Cosine similarity between the two sentences is 0.974

Original text is: Draconian laws made everybody miserable.

Simplified text is: the laws made everybody sick. .

Simplicity of original text is: 22.6

Simplicity of simplified text is : 28.25

Cosine similarity between the two sentences is 0.923

8.2 Detoxification

Given here are some outputs corresponding the inputs we fed into our model for detoxification:

Original text (non toxic): *oh yes evil christian missionaries many slaughtered indians lost lives due disease starvation etc lumped genocidists mention yes make word make stories.*

Detoxified and Paraphrased: yes, evil christian missionaries many killed indians due to disease starvation, etc lumped together.

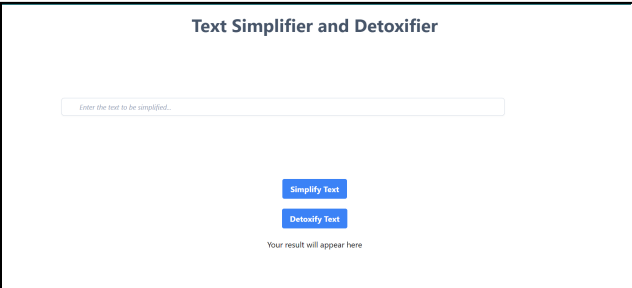
Original text (toxic): *black racist crap still gm website stopped talking black people long ago afraid life since 8090 crime black people kinda makes sense it oh wait thats profiling driving black know leftist marxists blm nutjobs change names fast one keep up facts speak like this wash away letting identity marxists spew racist hatred white people think not commit crimes world f would give time day clean sht first people problem.*

Detoxified and Paraphrased: A black racist deterrent has been on the internet long before black people's life since the 1980s, but black people's profiling has made it seem that this washout is not about tolerating bigotry.

Original text (toxic): *Fuck you bitch*

Detoxified and Paraphrased: You idiot.

8.3 A Basic Frontend



The screenshot shows a web application titled "Text Simplifier and Detoxifier". It features a text input field with the placeholder "Enter the text to be simplified.". Below the input field are two buttons: "Simplify Text" and "Detoxify Text". At the bottom of the interface, there is a label that says "Your result will appear here".

We have designed a preliminary user interface and merged it with the backend system that contains the simplification and detoxification models. The interface consists of an input field to receive text input and two buttons to initiate the desired action, either simplification or detoxification. The frontend is developed using ReactJS and Tailwind, while the backend comprises Flask and Python.

9. References

[1] Hironson, "hatesonar: Hate Speech Detection Library for Python.". Available: <https://github.com/Hironson/HateSonar>

- [2] S. Zannettou, M. Elshrief, E. Belding, S. Nilizadeh, and G. Stringhini, "Measuring and Characterizing Hate Speech on News Websites," in 12th ACM Conference on Web Science, Southampton United Kingdom, Jul. 2020, pp. 125–134. doi: 10.1145/3394231.3397902.
- [3] "Perspective API - How it works." <https://perspectiveapi.com/how-it-works/>.
- [4] H. Hosseini, S. Kannan, B. Zhang, and R. Poovendran, "Deceiving Google's Perspective API Built for Detecting Toxic Comments." arXiv, Feb. 26, 2017. doi: 10.48550/arXiv.1702.08138.
- [5] "Rewordify.com | Understand what you read." <https://rewordify.com/index.php>.
- [6] "Rewordify.com | Help Center." <https://rewordify.com/helpwhatcannotdo.php>.
- [7] P. Athanasiadou, G. Andreou, and E. Gana, "ICT and specific learning disabilities: A proposition for the use of the software Rewordify in the foreign language learning by students with reading comprehension difficulties," *Διεθνές Συνέδριο Για Την Ανοικτή Εξ Αποστάσεως Εκπαίδευση*, vol. 10, no. 3A, Art. no. 3A, 2019, doi: 10.12681/icodl.2298.
- [8] "Need to simplify/summarize text online?" <https://www.simplish.org/>.
- [9] D. Moskovskiy, D. Dementieva, and A. Panchenko, "Exploring Cross-lingual Text Detoxification with Large Multilingual Language Models," in Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, Dublin, Ireland, May 2022, pp. 346–354. doi: 10.18653/v1/2022.acl-srw.26.
- [10] A. Xu, E. Pathak, E. Wallace, S. Gururangan, M. Sap, and D. Klein, "Detoxifying Language Models Risks Marginalizing Minority Voices." arXiv, Apr. 13, 2021. doi: 10.48550/arXiv.2104.06390.
- [11] N. H. Qasmi, H. B. Zia, A. Athar, and A. A. Raza, "SimplifyUR: Unsupervised Lexical Text Simplification for Urdu," in Proceedings of the Twelfth Language Resources and Evaluation Conference, Marseille, France, May 2020, pp. 3484–3489. [Online]. Available: <https://aclanthology.org/2020.lrec-1.428>
- [12] S. Stajner, "Automatic Text Simplification for Social Good: Progress and Challenges," in Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online, Aug. 2021, pp. 2637–2652. doi: 10.18653/v1/2021.findings-acl.233.
- [13] Laugier et al., "Civil Rephrases Of Toxic Texts With Self-Supervised Transformers."
- [14] David Dale, Anton Voronov, Daryna Dementieva, Varvara Logacheva, Olga Kozlova, Nikita Semenov, and Alexander Panchenko. 2021. Text Detoxification using Large Pre-trained Neural Models. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 7979–7996, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.