CSE508: Information Retrieval

LSD for Indian Languages :Lexical Text Simplification and Detoxification

Aanya Trehan	Abhimanyu Bhatnagar	Aniket Goel	Apoorva Arya	Naman Kaushik	Sejal Kardam
2020419	2020273	2020281	2020032	2020088	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. Literature Review

2.1 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.

2.2 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.

2.3 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.

2.4 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].

2.5 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.

2.6 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].

2.7 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 propose to solve this problem in three parts:
4.1.1 Identifying the complex words in the sentence: 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 Masking of the identified word and find potential replacements

For this we first masked the identified complex word and then utilized the MLM (Masked language modeling) provided by BERT (by HuggingFace) to predict the most appropriate words in place of the masked word. The model provides a list of the most appropriate words that can come in place of the masked word. This masking was done for each complex word found in a sentence.

4.1.3 Choosing the most appropriate replacement for the complex word

In the previous step, we obtained a set of words to replace each complex word. In order to choose the most appropriate word, the complex word was replaced with one of the proposed new words in the sentence to obtain the simplified sentence. Then the similarity between the original sentence and the modified sentence was found using cosine_similarity. The most appropriate replacement word chosen was the one in which the cosine similarity between the original and the modified sentence was maximum.

4.1.3 Evaluation Method

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.

4.2 Detoxification

As part of our baseline model, we developed a model that performs two tasks: (1) it detects whether a sentence or comment contains toxic content, and (2) it replaces the toxic words in the comment with asterisks to make it suitable for readers of all ages. To classify the text, we employed a machine learning algorithm that uses word vector embeddings generated by Word2Vec. We experimented with several ML models and the Jigsaw dataset was used for training the model. For the second task, we used a dataset containing a list of toxic words to mask their appearance in our comments and text.

The word2vec embeddings that were generated serve as input features for training a logistic regression model on the pre-processed text data. The model was able to achieve an accuracy of 91% on the dataset, which serves as a baseline for future improvements. Additionally, the dataset used was comprehensive and effectively replaces all toxic words, resulting in "detoxified" sentences that are appropriate for all audiences.

Using the baseline model as a threshold, the team plans to explore text generation techniques for detoxification in order to develop a more comprehensive solution and surpass the baseline scores. This is a relatively unexplored area of research, and the team hopes to bridge this gap by integrating text generation for detoxification and the simplification task as one whole unit. We finally intend to use both these models, one after another on a given text.

5. Results:

5.1 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 plan or i will torture you .
Simplicity of original text is: 53.75
Simplicity of simplified text is : 55.47

Original text is: The door to his residence was locked
Simplified text is: the door to his apartment was locked
Simplicity of original text is: 42.36
Simplicity of simplified text is : 42.65

Original text is: Draconian laws made everybody miserable.
Simplified text is: the laws made everybody insane .
Simplicity of original text is: 22.6
Simplicity of simplified text is : 27.78

5.2 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: oh yes evil christian missionaries many slaughtered indians lost lives due disease starvation etc lumped genocidists mention yes make word make stories.

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

5.3 A Basic Frontend

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.

6. Proposed Solutions for Future

6.1 Lexical text simplification

The given model manages to simplify the sentences, but sometimes this leads to deviation from the meaning of the original sentence. We intend to improve upon this. One way to do this is - Out of the candidate words generated from BERT, the current scheme replaces masked words in the sentence with a probable predicted word one by one. We could replace a combination of masked words together with probable predicted words and then measure the similarity between the original sentence and the new sentence.
Also while choosing the replacement word, we could also take into consideration the zipf value of all the possible words so the resulting text is simpler.

6.2 Detoxification

In our present study, we have developed an improved model for detecting and replacing harmful or offensive phrases in text using a more expansive vocabulary. However, our future work aims to take this approach to the next level by building a sophisticated text generation model that is capable of effectively and consistently generating neutral or positive alternatives to such phrases. This involves utilizing advanced techniques such as natural language processing (NLP), automatic thesaurus generation, and context-aware text generation. Our proposed approach will not only detoxify the text but also ensure that the overall meaning and sentiment are

preserved. Moreover, we intend to apply this advanced model to Indian languages, which would require the availability of sufficient linguistic resources and time. Through this work, we aim to contribute to the development of novel and effective approaches to improve the quality and inclusiveness of text for a diverse range of audiences.

7. References

[1] Hironsan, “hatesonar: Hate Speech Detection Library for Python.”. Available: https://github.com/Hironsan/HateSonar

[2] S. Zannettou, M. Elsherief, 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.”