

¹Almas Namazbayev

¹Suleyman Demirel University, Kaskelen, Kazakhstan

*e-mail: 221107042@stu.sdu.edu.kz

ANALYSIS OF NLP METHODS TO IDENTIFY OFFENSIVE LANGUAGE

Abstract: This research focuses on the application of Natural Language Processing (NLP) techniques to detect offensive language in textual data aimed at improving content moderation on digital communication platforms. Using a dataset, the study evaluates the effectiveness of advanced NLP models and algorithms in detecting explicit and implicit forms of offensive language. The core of the analysis centers around transformer-based models, in particular BERT (Bidirectional Encoder Representations from Transformers). The study addresses the challenges of offensive expression detection, highlighting both the successes and challenges faced in accurately classifying text as offensive or not. This research contributes to ongoing efforts to create a safer and more inclusive digital environment by offering insight into the potential of NLP technologies to address the widespread problem of profanity on the Internet.

Keywords: Natural Language Processing (NLP), Offensive Language Detection, Bidirectional Encoder Representations from Transformers (BERT), Text Classification, Explicit and Implicit Offensiveness, Algorithm Effectiveness

1. Introduction

This exploration delves into the complex realm of Natural Language Processing (NLP) techniques designed for the identification of offensive language within textual data. Offensive language, a widespread issue in digital communication, spans a broad array of expressions aimed at harming, insulting, or provoking individuals or groups. Grasping the subtleties of offensive language is pivotal for devising effective moderation strategies and cultivating safer online spaces.

Various forms of offensive language emerge, each with distinct characteristics and implications. These include:

- **Explicit Offensiveness:** Characterized by direct and overt use of profanity, slurs, or derogatory terms targeting specific individuals or marginalized groups, such as racial slurs, sexist comments, and hate speech.
- **Implicit Offensiveness:** Relies on subtlety or ambiguity to convey derogatory or discriminatory messages, with sarcasm, innuendo, and microaggressions necessitating nuanced analysis for detection.[1]

- **Passive-Aggressive Language:** Manifests through indirect expressions of hostility, resentment, or frustration, often cloaked in superficial politeness or feigned innocence.[2]
- **Stereotyping and Generalizations:** Offensive language that perpetuates stereotypes or makes broad generalizations about individuals or groups based on their characteristics, identity, or affiliations, including racial stereotypes and gender-based assumptions.[3]
- **Sexual Harassment and Inappropriate Content:** Consists of sexually explicit content, innuendos, and unwelcome advances, creating hostile or uncomfortable environments and infringing on individuals' rights to safety and respect.[4]

2. Literature Review

The advancement in offensive language detection reflects a concerted effort across multiple disciplines, employing a range of Natural Language Processing (NLP) and Machine Learning (ML) strategies. This section delves into the nuanced methodologies that have shaped the field, highlighting both the technological progress and the ethical frameworks guiding this research.

2.1 Key Studies:

1. **Toxic Spans Detection:** This study underscored the effectiveness of sequence labeling and rationale extraction in pinpointing toxic content within texts, showcasing the importance of detailed content analysis.[5]
2. **ConvAbuse:** Investigated the specific challenges of detecting nuanced abusive language directed at AI systems, underlining the need for refined detection mechanisms in digital interactions.[6]
3. **Detection in Tweets Using Deep Learning:** Illustrated the impact of incorporating user behavioral data into RNN classifiers, significantly improving the identification of racist and sexist content on social media.[7]
4. **Offensive Content Detection in Low-Resource Languages:** Demonstrated the potential of BiLSTM networks and transfer learning to overcome data limitations in low-resource language contexts, setting a precedent for future research.[8]
5. **ToxiSpanSE:** Introduced an innovative model capable of providing explainable insights into toxic content within software engineering discussions, particularly in code review comments.[9]
6. **Ethical and Human Rights Perspective:** Stressed the necessity of integrating ethical and human rights principles into the development of detection technologies, advocating for approaches that respect individual rights and promote societal values.[10]
7. **Fake News Detection Review:** Conducted a comprehensive evaluation of NLP and ML methods for fake news detection, highlighting the superiority of Ensemble Methods in tackling misinformation.[11]

8. Abusive Language Detection Review: Aggregated various strategies for identifying abusive language on platforms like Twitter, emphasizing the critical role of extensive resource development.[12]
9. BERT-Based Models for Offensive Language Detection: Explored enhancements in BERT models through the customization of attention probabilities, achieving notable success in detecting offensive content in English and Persian.[13]
10. Offensive Language Identification in Low Resource Languages: Addressed the challenges of detecting offensive language in the Kazakh language, illustrating the effectiveness of BiLSTM networks in resource-constrained environments.[14]

2.2 Approaches and Techniques:

Research in this domain has significantly benefited from the deployment of deep learning technologies such as Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and the Bidirectional Encoder Representations from Transformers (BERT) model. These models excel in decoding the complexities of language, offering nuanced insights into the detection process. Notably, advancements have been propelled by customizing attention mechanisms within these models and integrating analyses of user behavior, which have proven instrumental in enhancing detection accuracy. Furthermore, the development and meticulous curation of datasets, particularly for underrepresented languages, have been pivotal in refining model performance and ensuring broader applicability.

3. Research Methods

3.1 Dataset:

The research utilized the 'comments.csv' dataset, which was sourced from public internet data. This dataset, available at https://github.com/ipavlopoulos/toxic_spans. It provides a resource for applying and evaluating advanced NLP models and algorithms in the context of offensive language detection.

In search of effective methods for identifying toxic fragments in the text, our study uses an approach to labeling sequences used by SemEval-2021 participants. This approach uses the BERT model to analyze and classify text data. To train the model, we use a dataset consisting of 10,629 annotated messages, each of which is labeled for toxicity.

| Comment ID | Comment Text |
|------------|---|
| 239607 | Yet call out all Muslims for the acts of a few will get you pilloried. So why is it okay to smear an entire religion over these few idiots? Or is this because it's okay to bash Christian sects? |
| 239612 | This bitch is nuts. Who would read a book by a woman. |
| 240311 | You're an idiot. |
| 240400 | Nincompoop, that's a nice one! I'm partial to silly goose. |
| 240941 | I honestly cannot decide if these guys are complete morons or the most patriotic heroes this country has seen in a long time. |

Table 1: Excerpt from the "comments.csv" dataset

3.2 Data Preprocessing:

In preparing the dataset for analysis, this study engaged in comprehensive data preprocessing, conducted on Google Colab to leverage its extensive computing capabilities. The preprocessing included:

1. Elimination of Special Characters: Utilizing Python's regex library, the research filtered out non-alphanumeric symbols from the dataset to ensure a focus on textual information.
2. Text Normalization: The process standardized text to a uniform case, applying the NLTK library for managing textual inconsistencies, including the expansion of contractions.
3. Tokenization: This phase involved dissecting the cleaned text into its basic units or tokens using NLTK, facilitating granular linguistic analysis.
4. Removal of Stopwords: To concentrate on significant textual elements, common words of minimal analytical value were removed with the aid of NLTK's comprehensive stopwords list.

By performing these preprocessing steps, the dataset is cleaned, standardized, and ready for further analysis, ensuring the accuracy and reliability of subsequent modeling tasks.

```
def preprocess_text(text):
    # Removing special characters and numbers, reducing them to lowercase
    text_clean = re.sub(r'^a-zA-Z\s', '', text).lower()

    # Tokenization
    tokens = word_tokenize(text_clean)

    # Deleting stop words
    tokens = [token for token in tokens if token not in stop_words]

    return ' '.join(tokens)
```

Figure 1. The text preprocessing function is used to prepare text data for further analysis. It removes special characters, numbers, lowercase translates text, splits text into tokens and removes stop words.

3.2.1 Integrating a list of offensive words:

In addition to the basic preprocessing steps, it included a list of offensive words for the purposes of initial detection. It is stored in a text file named "offensive_words.txt " and contains predefined offensive terms. A function named is_offensive was designed to iterate through this list and identify comments containing these words. Comments marked as offensive were appropriately labeled in the dataset.

```
with open('offensive_words.txt', 'r') as file:
    offensive_words = file.read().splitlines()

def is_offensive(comment):
    for word in offensive_words:
        if word in comment.lower():
            return 'Offensive'
    return 'Inoffensive'

comments_df['offensive'] = comments_df['processed_text'].apply(is_offensive)
print(comments_df[['processed_text', 'offensive']].head())
```

| | processed_text | offensive |
|---|---|-----------|
| 0 | yet call muslims acts get pilloried okay smear... | Offensive |
| 1 | bitch nuts would read book woman | Offensive |
| 2 | youre idiot | Offensive |
| 3 | nincompoop thats nice one im partial silly goose | Offensive |
| 4 | testing purposes idiot cant stand ignorant don... | Offensive |

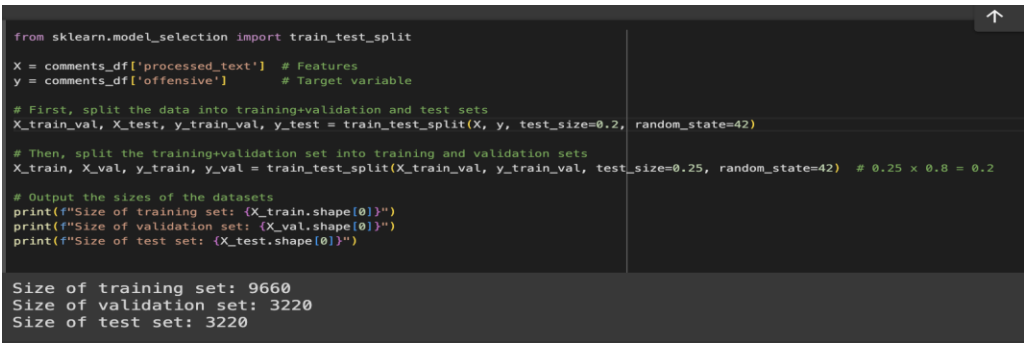
Figure 2. Offensive Word List Integration

3.2.2 Data separation

Data separation into training, validation and test samples. The goal is to divide the data into three subsets for training, validation and testing of the model. Explanation of the code:

- The sklearn.model_selection library is imported to separate the data.
- The attributes (X) and the target variable (y) are determined from the dataframe.

- The data is first divided into training/test and validation samples (80%/20%) using `train_test_split`.
- The training/validation sample is then divided into training and validation (75%/25%) using `train_test_split`.



```

from sklearn.model_selection import train_test_split

X = comments_df['processed_text'] # Features
y = comments_df['offensive']      # Target variable

# First, split the data into training+validation and test sets
X_train_val, X_test, y_train_val, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Then, split the training+validation set into training and validation sets
X_train, X_val, y_train, y_val = train_test_split(X_train_val, y_train_val, test_size=0.25, random_state=42) # 0.25 x 0.8 = 0.2

# Output the sizes of the datasets
print("Size of training set: {X_train.shape[0]}")
print("Size of validation set: {X_val.shape[0]}")
print("Size of test set: {X_test.shape[0]}")

```

Size of training set: 9660
Size of validation set: 3220
Size of test set: 3220

- The sizes of all samples are displayed for information.

Figure 3. Data separation into training, validation and test samples.

3.3 Model Selection:

For the analysis of offensive language, state-of-the-art Natural Language Processing (NLP) models and algorithms were selected to address the complexity of text data. Among the primary models chosen was BERT (Bidirectional Encoder Representations from Transformers), a transformer-based model renowned for its exceptional performance in various NLP tasks, including offensive language detection. BERT's bidirectional architecture enables it to capture intricate contextual relationships within text, making it particularly adept at discerning the nuances of offensive language.

4. Research Results

In the exploration of Natural Language Processing (NLP) techniques for identifying offensive language within textual data, a dataset named "comments.csv," consisting of user comments, was utilized. This diverse collection offered a unique opportunity to apply and assess the effectiveness of advanced NLP models and algorithms in detecting various forms of offensive language.

In our study, we faced similar challenges as the participants of SemEval-2021, especially with regard to the difficulty of detecting toxicity at a detailed level. We have confirmed that detecting toxic fragments in a text is much more difficult than classifying entire posts as toxic or non-toxic, especially when it comes to implicit toxicity and the intricacies of language use.

4.1 Theoretical Insights:

The research was grounded in the sophisticated architecture of BERT (Bidirectional Encoder Representations from Transformers). Based on the transformer architecture, these models excel in capturing the context of words

```
# Receipt y_pred
y_pred = []
for comment in X_val:
    classification_results = classify_text(comment)
    y_pred.append(classification_results)

from sklearn.metrics import accuracy_score

# Assuming y_pred is a list of predictions from your model
y_pred_decoded = ["Offensive" if label[0]["label"] == "LABEL_1" else "Inoffensive" for label in y_pred]

# Calculating accuracy
accuracy = accuracy_score(y_true, y_pred_decoded)
print(f"Model accuracy: {accuracy}")

Model accuracy: 0.8962732919254658
```

within sentences, a capability crucial for the nuanced and context-dependent task of identifying offensive language.

4.2 Experimental Results:

After preprocessing the data — which included removing special characters, normalizing text, tokenizing, and eliminating stopwords — BERT and its variants were applied to classify comments into offensive and non-offensive categories. These models exhibited significant proficiency in identifying explicit offensiveness, such as profanity and slurs, with high accuracy, thanks to their ability to grasp the context and connotations of words. However, the task became more challenging when dealing with implicitly offensive language, passive-aggressive remarks, and subtler forms of harassment. In these instances, the deep contextual understanding of models akin to BERT was invaluable, yet it also underscored the necessity for further refinement in detecting linguistic subtleties and ambiguities.

In light of our objectives, the performance metrics obtained from the BERT-based model have been encouraging. For instance, one of the test outputs yielded a result of `{'label': 'LABEL_1', 'score': 0.7164639234542847}`, which indicates a moderate to high level of confidence in classifying comments under the specified category. This particular outcome, while not absolute, demonstrates a significant capability of the model to discern between offensive and non-offensive content with a degree of reliability. It highlights the potential and current limitations of using deep learning for nuanced text analysis, emphasizing the need for ongoing model training and fine-tuning to better capture the complexities of human language, especially in the context of implicit offensiveness or subtlety.

To evaluate this performance quantitatively, we employed a Python script. This script iterated through the validation set comments (`X_val`), utilized a `classify_text` function to generate predictions for each comment, and stored the results in the `y_pred` list. The `accuracy_score` function from the `sklearn.metrics`

library was then used to calculate the model's accuracy on the validation set, achieving a score of 0.8963.

Figure 4. Code for Model Evaluation.

4.3 Patterns in Offensive Language:

The analysis indicated that offensive language in the dataset was predominantly explicit, characterized by the use of profanity, slurs, and derogatory terms. Such language was relatively straightforward to detect due to its overt nature, underlining the persistent challenge of fostering civility and respect in online interactions.

4.4 Detected Patterns:

A recurring pattern of specific types of offensive expressions was observed, with certain derogatory terms and slurs frequently appearing. This recurrence highlights the persistence of prejudiced attitudes and biases within online environments. Moreover, the dataset contained instances of comments that, while not overtly offensive, could be interpreted as microaggressions, further complicating the detection task.

5. Conclusions

The study successfully demonstrated the potential of current NLP models, particularly BERT, in capturing the nuances of profanity. The transformer-based model demonstrated exceptional skill in recognizing explicit forms of insults, such as profanity and profanity, due to its deep contextual understanding capabilities. However, the study also revealed difficulties in dealing with implicitly offensive language, passive-aggressive comments and more subtle forms of harassment. This highlighted the need for further algorithm improvements to better recognize these complex communication patterns.

Several challenges arose throughout the research process, including the nuanced nature of offensive language, which often required complex interpretations of context and intent beyond simple word recognition. The variability of profanity, including not only outright insults but also more subtle, context-dependent expressions of contempt or aggression, posed significant obstacles to detection accuracy.

Looking ahead, the research aims to address these challenges by exploring more advanced machine learning techniques and incorporating a wider range of linguistic features into the analysis. Future work will focus on making the models more sensitive to subtleties of language, including irony, sarcasm, and cultural nuances, to improve detection of covert insults. It is also planned to expand the coverage of the dataset to include a wider range of languages and dialects, thereby improving the applicability of the models in different linguistic contexts. In addition, further research will focus on developing more robust

algorithms that can adapt to the changing landscape of online communication, ensuring that NLP techniques remain effective in the face of changing patterns of language use.

Building on the foundations laid by this study, future research will continue to advance the field of offensive language detection, contributing to the development of a safer and more inclusive online environment. The pursuit of more advanced NLP techniques and a deeper understanding of the complexities of human communication is a testament to the ongoing commitment to using technology to improve digital discourse.

References

- 1 Lewandowska-Tomaszczyk, B., Bączkowska, A., Liebeskind, C., Valunaite Oleskeviciene, G., & Žitnik, S. (2023). An integrated explicit and implicit offensive language taxonomy. *Lodz Papers in Pragmatics*, 19(1), 7-48.
- 2 Lim, Y. O., & Suh, K. H. (2022). Development and validation of a measure of passive aggression traits: the Passive Aggression Scale (PAS). *Behavioral Sciences*, 12(8), 273.
- 3 Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., & Kumar, R. (2019). Predicting the type and target of offensive posts in social media. *arXiv preprint arXiv:1902.09666*.
- 4 Oswal, N. (2021). Identifying and Categorizing Offensive Language in Social Media. *arXiv preprint arXiv:2104.04871*.
- 5 Pavlopoulos, J., Sorensen, J., Laugier, L., & Androutsopoulos, I. (2021, August). SemEval-2021 task 5: Toxic spans detection. In *Proceedings of the 15th international workshop on semantic evaluation (SemEval-2021)* (pp. 59-69).
- 6 Curry, A. C., Abercrombie, G., & Rieser, V. (2021). ConvAbuse: Data, analysis, and benchmarks for nuanced abuse detection in conversational AI. *arXiv preprint arXiv:2109.09483*.
- 7 Pitsilis, G. K., Ramampiaro, H., & Langseth, H. (2018). Detecting offensive language in tweets using deep learning. *arXiv preprint arXiv:1801.04433*.
- 8 Adam, F. M., Zandam, A. Y., & Inuwa-Dutse, I. (2023). Detection of Offensive and Threatening Online Content in a Low Resource Language. *arXiv preprint arXiv:2311.10541*.
- 9 Sarker, J., Sultana, S., Wilson, S. R., & Bosu, A. (2023, October). ToxiSpanSE: An Explainable Toxicity Detection in Code Review Comments. In *2023 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)* (pp. 1-12). IEEE.
- 10 Kiritchenko, S., Nejadgholi, I., & Fraser, K. C. (2021). Confronting abusive language online: A survey from the ethical and human rights perspective. *Journal of Artificial Intelligence Research*, 71, 431-478.

- 11 Hoy, N., & Koulouri, T. (2021). A systematic review on the detection of fake news articles. arXiv preprint arXiv:2110.11240.
- 12 Naseem, U., Khan, S. K., Farasat, M., & Ali, F. (2019). Abusive language detection: A comprehensive review. Indian Journal of Science Technology, 12(45), 1-13.
- 13 Alavi, P., Nikvand, P., & Shamsfard, M. (2021). Offensive Language Detection with BERT-based models, By Customizing Attention Probabilities. arXiv preprint arXiv:2110.05133.
- 14 Toktarova, A., Abushakhma, A., Adylbekova, E., Manapova, A., Kaldarova, B., Atayev, Y., ... & Aidarkhanova, A. (2023). Offensive Language Identification in Low Resource Languages using Bidirectional Long-Short-Term Memory Network. International Journal of Advanced Computer Science and Applications, 14(6).

¹Алмас Намазбаев

¹«SDU University», Қаскелең, Қазақстан

*e-mail: 221107042@stu.sdu.edu.kz

КЕМІТЕТІН СӨЙЛЕУДІ АНЫҚТАУ ҮШІН ТАБИҒИ ТІЛДІ ӨНДЕУ ӘДІСТЕРІН ТАЛДАУ

Андатпа: Зерттеу мәтіндік деректерде кемсітуші тілді анықтау үшін табиғи тілді өңдеу (NLP) әдістерінің қолданылуына бағытталған, бұл цифрлық байланыс платформаларында контентті модерациялауды жақсартуға бағытталған. Дерекқорды пайдалана отырып, зерттеу кемсітуші тілдің ашық және жасырын түрлерін анықтауда NLP-нің алдыңғы қатарлы модельдері мен алгоритмдерінің тиімділігін бағалайды. Талдаудың негізгі бөлігі трансформерлік модельдерге, атап айтқанда BERT (Bidirectional Encoder Representations from Transformers) моделіне арналған. Зерттеу кемсітуші сөйлеулерді анықтаудағы қиындықтарды қарастырады, мәтінді кемсітуші ретінде дәл жіктеудегі табыстар мен қиындықтарды атап өтеді. Бұл зерттеу интернетте кең таралған ашуланшақ сөздер мәселесін шешуде NLP технологияларының әлеуеті туралы түсінік бере отырып, қауіпсіз және қамтушы цифрлық ортаны құрудағы жалғасып жатқан ұмтылыстарға үлес қосады.

Кілт сөздер: Табиғи тілді өңдеу (NLP), Кемсітуші тілді анықтау, Трансформаторлардан Екі бағытты Кодтаушы Көріністер (BERT), Мәтінді жіктеу, Ашық және жасырын кемсітушілік, Алгоритмнің тиімділігі

¹Алмас Намазбаев

¹«SDU University», Қаскелең, Қазақстан

*e-mail: 221107042@stu.sdu.edu.kz

АНАЛИЗ МЕТОДОВ ОБРАБОТКИ ЕСТЕСТВЕННОГО ЯЗЫКА ДЛЯ ВЫЯВЛЕНИЯ ОСКОРБИТЕЛЬНОЙ РЕЧИ

Аннотация: Данное исследование сосредоточено на применении методов обработки естественного языка (NLP) для обнаружения оскорбительного языка в текстовых данных с целью улучшения модерации контента на цифровых платформах коммуникации. Используя набор данных, исследование оценивает эффективность передовых моделей и алгоритмов NLP в обнаружении явных и неявных форм оскорбительного языка. Основное внимание в анализе уделено моделям на основе трансформеров, в частности BERT (Bidirectional Encoder Representations from Transformers). Исследование рассматривает проблемы обнаружения оскорбительных высказываний, подчеркивая как успехи, так и трудности, с которыми столкнулись при точной классификации текста как оскорбительного или нет. Это исследование вносит вклад в продолжающиеся усилия по созданию более безопасной и инклюзивной цифровой среды, предоставляя представления о потенциале технологий NLP для решения широко распространенной проблемы нецензурной речи в Интернете.

Ключевые слова: Обработка естественного языка (NLP), Обнаружение оскорбительного языка, Представления двунаправленного кодера от трансформаторов (BERT), Классификация текста, Явная и неявная оскорбительность, Эффективность алгоритма