# Gender Classification on Twitter Based on Feeds and User Descriptions Using Xlnet-Fasttext

Vito Rozaan Alandeta[1], Derwin Suhartono[2]
[1]Computer Science Department, BINUS Graduate Program, Bina Nusantara University, Jakarta, Indonesia
[2]Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia
E-mail: vito.alandeta@binus.ac.id, dsuhartono@binus.edu

**Keywords:** gender classification, gender classification with transformers, xlnet, twitter classification, fasttext

**Received:** February 24, 2024

*This study proposes a gender classification method for Twitter data using a hybrid XLNet-fastText model. The objective is to enhance gender classification accuracy by leveraging the contextual understanding of XLNet and the semantic richness of fastText embeddings. Computational experiments were conducted on a dataset derived from Kaggle, focusing on user account descriptions and Twitter feeds. The proposed method achieved an accuracy of 71.4%, precision of 77%, recall of 60.9%, and F1-score of 68% for gender classification based on Twitter feeds. For user account descriptions, the scores were 72.4% for accuracy, 75.1% for precision, 63.24% for recall, and 68% for F1-score. These results outperform the baseline XLNet model and demonstrate the potential of the XLNet-fastText combination in improving text-based classification tasks. Our approach highlights a viable pathway to enhance gender classification on social media platforms, suggesting further improvements through multi-modal data integration.*

*Povzetek: Raziskava uporablja kombinacijo modelov XLNet in fastText za izboljšanje klasifikacije spola na podlagi Twitter objav in opisov uporabnikov.*

## 1 Introduction

The digital era has transformed human communication in unprecedented ways. Information and communication technology (ICT), particularly social media, has brought profound changes to how we engage, share information, and build social connections. Social media has become the primary platform for people to interact, exchange information, and expand their social networks. Social media is an online platform that enables users to create, share, and consume content in various formats. With billions of global users, it has become a crucial means of disseminating news, opinions, and fostering social relationships. The evolution of social media can be traced back to platforms like Friendster, MySpace, Facebook, Twitter, Instagram, and others, which have reshaped our communication methods.

However, the rise of social media has also given rise to various online scams and fraudulent activities, particularly romance scams. Figure 1 illustrates the growth of reports about romance scams over five years, showing that in 2021, total reported losses were more than six times what they were in 2017, and the number of reports grew to more than three times the 2017 number. These scams involve perpetrators creating fake online profiles with attractive photos and false identities, sometimes even using the identities of real people. They may study the information that people share online and pretend to share their interests. A notable case occurred in Jambi in 2022, where a perpetrator posed as a man and had an unregistered marriage with the victim, causing financial losses by borrowing money under false pretenses (tvonenews.com, 2022). Such cases highlight the need for robust mechanisms to detect and prevent online scams.

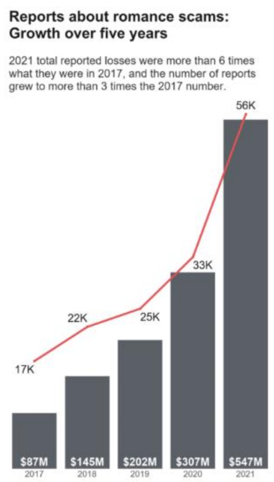


Figure 1: Number of reports about romance scams from 2017-2021 [1]

One potential solution is the application of natural language processing (NLP) techniques to classify and analyze text data on social media. Transformers is a natural language modeling (NLP) architecture that has changed the landscape of natural language processing since its introduction. Developed by Google Research in 2017, transformer brings significant innovation by replacing conventional approaches that use recurrent neural networks (RNN) or long short-term memory (LSTM). The advantages of transformers are not limited to text processing alone; This architecture has been successfully adopted in various fields such as computer vision and voice recognition. One of the well-known transformer models is BERT (Bidirectional Encoder Representations from Transformers) (Devlin, et al., 2019), which has solved various NLP tasks and became the basis for many subsequent NLP model developments. This transformation opens the door to new developments in artificial intelligence and plays a central role in improving the system's ability to understand and produce text with a higher degree of accuracy. Examples of the application of Transformer in carrying out text classification include research on automatic news summarization carried out by (Gupta, A., et al., 2022), analysis of public opinion carried out in research (Anwar, M. et al., 2021), and research on the classification of Covid19 disease taken from Twitter using XLNet and BERT carried out by (Kumar, D., et al., 2021).

Based on the problem described, this research focuses on designing an XLNet model and combining it with fastText as an additional embedding to optimize XLNet's performance in terms of accuracy, recall, precision, and F1-score. The research aims to carry out text-based gender classification using data from Twitter feeds and user descriptions. Similar research was conducted by (Onikoyi, B., et al., 2023) using a machine learning model combined with several embeddings.

## 2 Related works

Research related to Twitter-based gender classification has been carried out in previous years, providing a foundation for further studies in this field. For instance, (Movahedi Nia, et al., (2022)) focused on a multi-modal application using the transformer model and compared it with several other models. Multi-modal means combining two different types of data for classification. This experiment used the PAN-2018 dataset, which contains images and tweet text in various languages, including English, Spanish, and Arabic. The dataset was divided into 3000 Twitter user training data and 1900 Twitter user testing data. The methods used included BERT, FNN, XGBoost, Random Forest, SVM, and Naïve Bayes. The results obtained in this research showed that using Fine-Tuning BERT, the performance results in gender classification using only text achieved a score of 79.94%, using text and images achieved a score of 81.89%, and combining text and images achieved a score of 85.52%.

Another example is the research by (Vashisth, P., & Meehan, K., 2020) which focused on the application of several machine learning models supported by various types of word embeddings. The models used included Logistic Regression, Support Vector Machine, and Naïve Bayes. The types of word embeddings used included W2Vec, GloVe, and Bag of Words (TF-IDF). The dataset used in this research comprised 20,000 English language tweets, divided into 80% training data and 20% testing data. The highest accuracy score obtained was 57.14% with the LR model, which showed an improvement over the baseline approach of 53.65%.

(Soldevilla, I., & Flores, N., 2021) also conducted gender classification research using BERT, focusing on gender violence messages on social media platforms like Reddit and Twitter. The dataset included 113,910 Reddit posts and 30,377 Twitter posts labeled "Violence" and "non-violence." The results obtained were an Area Under the Curve (AUC) of 0.9603, an accuracy of 0.8909, sensitivity of 0.8826, and specificity of 0.8989.

(Hashempour, R., et al,. 2019) focused on gender classification in various languages, such as Portuguese, French, Dutch, English, German, and Italian. The models used included Logistic Regression and Feed-Forward Neural Network (FFNN) with Inter-Language and Cross-Language settings. The dataset used the TwiSty corpus, containing gender annotations from 6482 authors. The Logistic Regression model achieved the highest score in the Inter-language setting with 70.30 in French, while FFNN achieved the highest score in the Cross-language setting with 85.62 in Italian using a 3-layer FFNN.

(Puertas, E., et al., 2019) conducted research on classifying bot accounts using a dual-language dataset in English and Spanish. The English dataset consisted of 4,120 data points, while the Spanish dataset consisted of 1,500 tweet data points. The research used various machine learning models and achieved the best results with the Random Forest classifier for bot classification, obtaining a macro F1 score of 91% for English and 89% for Spanish. For gender classification, the Random Forest classifier achieved the best results for English with a macro F1 score of 81%, while the logistic regression classifier achieved a macro F1 score of 75% for Spanish.

(Staykovski, T., 2019) focused on predicting gender from tweet text and images using a stack technique consisting of two main parts: predicting gender from text based on n-grams and TF-IDF and predicting gender from images using different layers of classifiers. The dataset consisted of English language data with 4,120 files, each containing 100 tweets from unique authors. The results showed that TF-IDF achieved an accuracy of 83.62%, Doc2Vec achieved 83.22%, and combining both achieved 85.96%.

(Saeed, U., & Shirazi, F., 2019) used Multinomial Naïve Bayes for bot account classification and Decision Tree for gender classification of Twitter users. The dataset used was PAN-2019, containing 412,000 labeled tweets, divided into 288,000 training tweets and 124,000 testing tweets. The results showed that MultinomialNB achieved an accuracy score of 79.51% for bot classification, while Decision Tree achieved 56.55% for gender classification.

(Ouni, S., et al., 2022) used a similar dataset but applied it in two languages, English and Spanish. The models used included Logistic Regression, SVM RBF, Naïve Bayes, SVM Linear, Random Forest, and CNN. The Random Forest model achieved the highest accuracy for bot and gender classification in both languages, with 93.06% and 90.04% for English, and 90.53% and 89.11% for Spanish, respectively.

(Alroobaea et al., 2020) classified the age and gender of Twitter users using machine learning and deep learning models. The machine learning models included SVM, Random Forest, K-NN, Decision Tree, and Naïve Bayes, while the deep learning models included CNN and LSTM. The dataset used was PAN-AP-2019, containing 14,166 English language tweets. The results showed that the best machine learning model for age classification was SVM with an accuracy score of 66%, while CNN and LSTM achieved 74% and 72%, respectively. For gender classification, Naïve Bayes, SVM, Neural Network, and Random Forest achieved an accuracy score of 69%, while the deep learning model achieved 99%.

Although machine learning and deep learning have become popular methods in text classification, word embedding also has a crucial role in improving its quality and effectiveness. By using word embedding, we can capture the semantic relationships between words in a text. This allows the model to better understand the context and meaning of words, enriching the text representation and reducing the dimensionality of the data. Therefore, the use of word embedding together with machine learning and deep learning is a powerful combination to improve text classification performance and make models better able to handle the complexity of natural language.

Table 1: Related works summary

| Study | Models | Dataset | Performance Matrix |
|---|---|---|---|
| (Movahedi Nia, et al., 2022), | BERT, FNN, XGBoost, Random Forest, SVM, Naïve Bayes | **PAN-2018** | **Text**: 79.94%, **Text+Image**: 85.52% |
| (Vashisth, P., & Meehan, K., 2020). | Logistic Regression, SVM, Naïve Bayes | **Kaggle** 20,000 Tweets | **LR**: 57.14%, **Baseline**: 53.65% |
| Puertas, E., et al., 2019 | Random Forest, Logistic Regression | **PAN-2019** 4120 English tweets & 1500 Spanish tweets | **Bot**: **RF(EN)**: 91%, **RF(ES)**: 89%; **Gender**: **RF(EN)**: 81%, **LR(ES)**: 75% |
| Staykovski, T. (2019) | TF-IDF, Doc2Vec | **PAN-2019** | **TF-IDF** = 83.62%, **Doc2Vec** = 83.22%. **Combined** = 85.96%. |
| Alroobaea, R., et al., (2020) | SVM, Random Forest, K-NN, Decision Tree, Naïve Bayes, CNN, LSTM | **PAN-AP-2019** | **Age**: **SVM**: 66%, **CNN**: 74%, **LSTM**: 72%; **Gender**: **ML**: 69%, **DL**: 99% |
| Saeed, U., & Shirazi, F. (2019) | Multinomial Naïve Bayes, Decision Tree | **PAN-2019** | **Bot**: **MNB**: 79.51%, **Gender**: **DT**: 56.55% |
| Ouni, S., et al., (2022) | Logistic Regression, SVM RBF, Naïve Bayes, SVM Linear, Random Forest, CNN | **PAN-2019** English and Spanish | **Bot (EN)**: 93.06%, **Gender (EN)**: 90.04%, **Bot (ES)**: 90.53%, **Gender (ES)**: 89.11% |
| Soldevilla, I., & Flores, N. (2021) | BERT | 113,910 **reddits** and 30,377 **tweets** | **AUC**: 0.9603, **Accuracy**: 0.8909 |
| Hashempour, R., et al., (2019) | Logistic Regression, FFNN | **TwiSty Corpus** <br> **Inter-Language**: (Portuguese, French, Dutch, English, German, and Italian) <br> **Cross-Language**: (German, Italian) | **LR (Inter-Language): ** 70.30 **(French), FFNN** <br> **(Cross-Language):** 85.62 **(Italian)** |

# 3 Proposed method

The main objective of this research is to solve the problem of gender classification based on tweets and user descriptions and will be do some test for XLNet in gender classification using or without fastText, both of which will differentiate between male and female classes. Previously, we had directly tested both versions simultaneously, but in the end, we decided to do it one by one for one type of labelling. The research frameworks are shown in Figure 2.

## 3.1 Dataset

Dataset used in this research is taken from Kaggle written by Figure Eight that contains of more than 20000 English tweets with 26 columns directly taken from Twitter, all dataset has been labeled by the authors with four classes in gender column, there are male, female, brand, and unknown. [25]

Before the dataset used in further research, all unnecessary label such as brand and unknown will be removed and leaving only two active labels (Male and Female). The amount data has changed from 20050 into 11194 tweets.

Table 2: Amount all data labels before dropping unnecessary labels

| Twitter user gender classification | |
|---|---|
| Gender | Female (5725),<br>Male (5469),<br>Brand (4328),<br>Unknown (702) |

Once removing all brand and unknown tweets, so the amount of dataset that can be used is 11194 tweets and descriptions. The next step is data cleaning like removing unnecessary words such as https://, empty cells, numbers and symbols, remove emojis, and lowercase all text data.

11000 random tweets and descriptions have been selected from the dataset that are different from the training set and test set data used for the XLNet and fastText models. The model ran through the input of this global model to obtain the predictions computed by the concatenation embeddings between XLNet and fastText models.

Table 3: Amount all data labels after dropping unnecessary labels

| Twitter user gender classification | |
|---|---|
| Gender | Female (5725), Male (5469) |

After cleaning the dataset, next we augmenting the data using random insertion method with ROBeRTa Insertion and splitting the dataset into train and valid.

Table 4: Total data before & after augmentation

| | |
|---|---|
| Before augmentation | 11194 |
| After augmentation | 51296 |

As shown in Table 4. The augmentation inserting 40,102 random datasets, the reason doing the augmentation is the lack of transformers model that needed a lot of data for optimizing the training.

## 3.2 Data cleaning

The data cleaning involved removing URLs, emojis, and converting text to lowercase.

- **Removing URLs**: All URLs within the tweets were identified and removed to prevent irrelevant information from affecting the model's performance.
- **Removing emojis**: Emojis were removed as they can introduce noise and are not useful for text-based gender classification.
- **Converting text to lowercase**: To ensure uniformity and reduce the dimensionality of the text data, all text was converted to lowercase.

## 3.3 Tokenization

Tokenization is the process of breaking down a text into individual units, which can be words, phrases, symbols, or other meaningful elements. These units are called tokens. Tokenization is a fundamental step in natural language processing (NLP) and is essential for various text analysis tasks.

In the context of tokenization in NLP, a token can be as small as an individual word or as large as an entire sentence. The choice of tokenization unit depends on the specific task or analysis being performed.

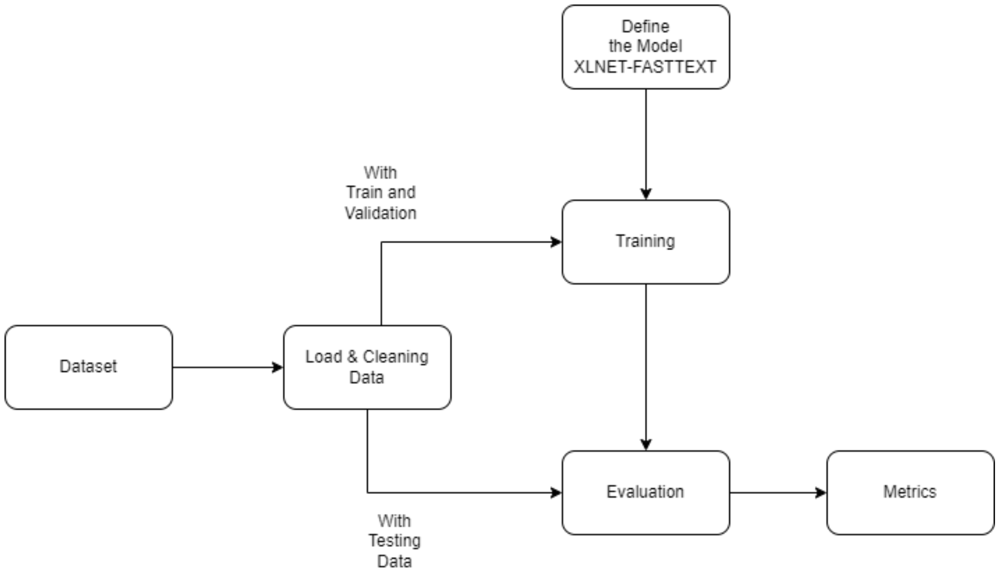


Figure 2: Research frameworks

Tokenization was performed using XLNet's built-in tokenizer. This step involves breaking down the text into individual tokens (words or subwords) that XLNet can process. XLNet's tokenizer is specifically designed to handle complex word structures and maintain context.

## 3.4 Handling special characters and stopwords

Special characters and stopwords (commonly used words that do not contribute significant meaning, such as 'and', 'the', etc.) were removed to reduce noise and improve model performance.

## 3.5 Word embedding with fasttext

Word embedding is a type of representation of words in a vector space that captures semantic relationships between words. In other words, it is a mathematical way to represent words so that similar words are closer to each other in the vector space. This representation is often used in natural language processing (NLP) and machine learning tasks.

The idea behind word embeddings is to convert words into numerical vectors in a way that preserves their semantic relationships. This allows machine learning models to better understand the meaning of words and the relationships between them. Word embeddings are particularly useful in tasks such as language translation, sentiment analysis, and text classification.

FastText is an open-source, free, lightweight library developed by Facebook's AI Research (FAIR) lab for efficient text classification and representation. It was designed to handle large text datasets efficiently and is particularly well-suited for tasks such as text classification, language identification, and word embedding.

FastText can generate continuous vector representations (embeddings) for words in each text. These embeddings capture semantic information and are useful for various natural language processing (NLP) tasks. fastText supports the training of text classifiers using a shallow neural network. This makes it effective for tasks where labeled data is available for training, such as sentiment analysis or topic classification.

The fastText algorithm is based on a neural network architecture that incorporates the Bag of Words (BoW) model and the subword information. The formula can be expressed as shown in formula (1).

$$-\frac{1}{N}\sum_{n=1}^{N} yn\ log(f(BAxn)) \qquad (1)$$

Where is the normalized bag of features of the nth document, the label, A and B the weight matrices. This model is trained asynchronously on multiple CPUs using stochastic gradient descent and a linearly decaying learning rate.

The training process involves updating the parameters of the neural network to minimize this objective function. fastText uses techniques like hierarchical softmax and negative sampling to make the training process more efficient. It's important to note that while this gives a general overview, the actual implementation details, hyperparameters, and optimizations may vary based on the specific version and settings used in fastText.

## 3.6 Proposed XLNet architecture

Figure 5 and Figure 6 show the XLNet architecture used in XLNet + fastText and vanilla XLNet. XLNet architecture consists of 7 layers, there are:

- Input Layer:

The diagram starts with the "Input Layer" represented by the input node. It represents the input text that is fed into the XLNet model for processing.

- Segment Embeddings:

The input text is passed through the "Segment Embeddings" layer, represented by the segment node. This layer assigns different embeddings to different segments or parts of the input text.

- Position Embeddings:

The input text is also passed through the "Position Embeddings" layer, represented by the position node. This layer assigns embeddings based on the position or order of the words in the input text.

- Attention Layer:

The segment embeddings and position embeddings are combined and fed into the "Attention Layer," represented by the attention node. The attention layer performs self-attention, allowing the model to focus on different parts of the input text while considering the dependencies between words.

- Feed-Forward Layer:

The output from the attention layer is passed through the "Feed-Forward Layer," represented by the feed-forward node. This layer applies a neural network with multiple layers and nonlinear transformations to capture complex patterns in the data.

- Residual Connections and Layer Normalization:

To facilitate better information flow and mitigate the vanishing gradient problem, "Residual Connections" are added between the attention layer and feed-forward layer. The "Add &; Layer Norm" operations, represented by add_norm_1 and add_norm_2, respectively, combine the output of the previous layer with its input and apply layer normalization.

- Output Layer:

Finally, the output from the feed-forward layer passes through the "Output Layer," represented by the output node, to generate the final output of the XLNet model.

The XLNet for fastText shown in Figure 4 has difference than the other from Figure 5 in terms of architecture form. This one has additional fastText layer, while the rest of the architecture is similar in terms of layers used in both architectures.

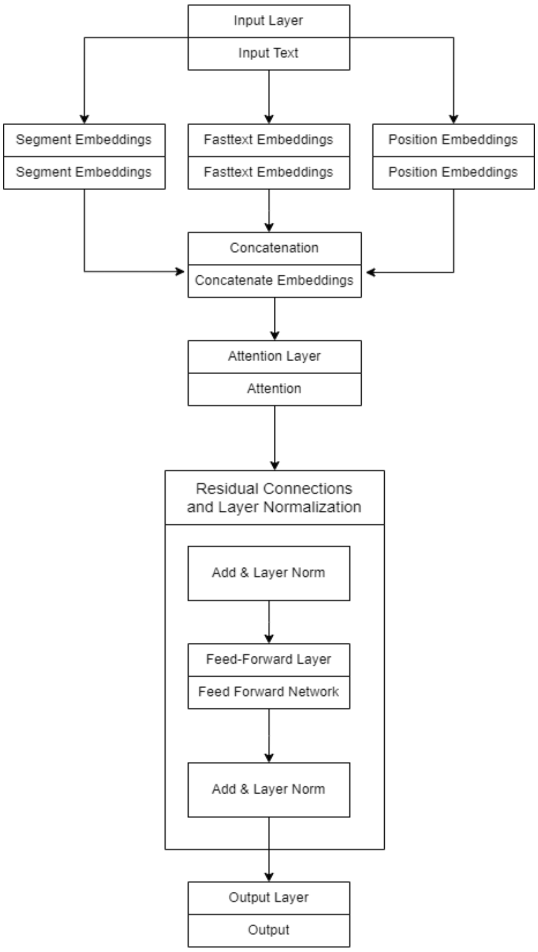


Figure 3: Vanilla XLNet architecture

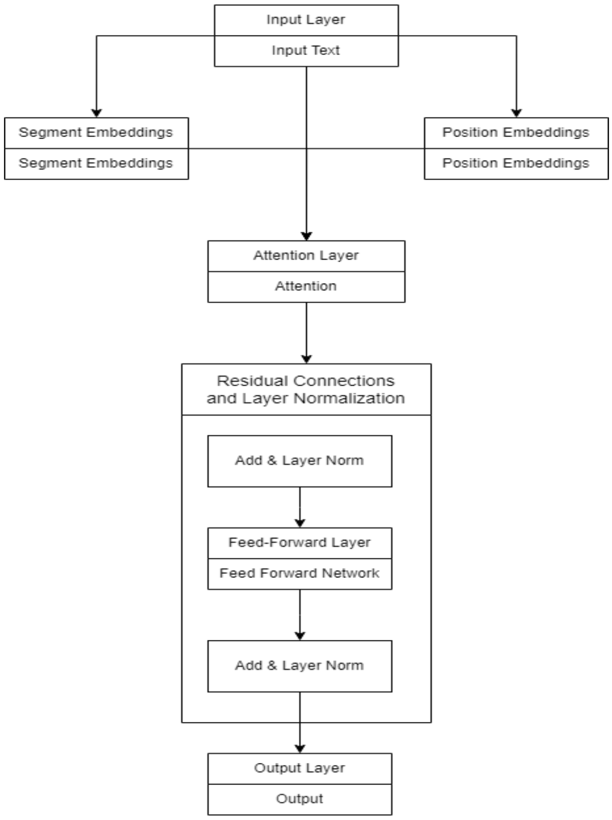


Figure 4: Proposed XLNet Architecture using fastText word embedding

### 3.7 Proposed XLNet + fastText architecture

All hyperparameter value used for pre-trained in both models is provided in Table 5.

Table 5: Hyperparameter tuning for both models

| Hyperparameters | Value |
|---|---|
| MAX LEN | 200 |
| Batch Size | 32 |
| Learning rate | 2e-5 |
| Optimizer | Adam epsilon |
| Epoch | 10 |

## 4 Experiment

Two treatments were equally applied to the proposed architecture models. The first treatment involves utilizing text and label data, specifically Twitter feeds, to evaluate the models' performance in classifying gender. The second treatment uses user account descriptions with the same gender labels to assess the models' ability to classify gender based on user descriptions.

Both treatments followed uniform procedures. The dataset was divided into training, validation, and test sets using an 80-10-10 split ratio, ensuring a comprehensive evaluation of the models. A batch size of 32 was utilized for all experiments, and the Adam optimizer was implemented with a learning rate ranging from 2e-5 to 3e-5 and an epsilon value of 1e-3. The models were trained for a maximum of 10 epochs. Early stopping was employed as a regularization technique to prevent overfitting, configured to halt training if the model's accuracy did not increase for three consecutive epochs.

This approach ensured that the training process was efficient and that the models did not overfit to the training data.

## 5 Result and discussion

This section focuses on the results, comparing the baseline model (XLNet) and the proposed model (XLNet + fastText) for text-based gender classification. The objective is to evaluate and compare their performance in classifying gender based on tweets and user descriptions into two classes: male and female. The evaluation metrics used are precision, recall, F1-score, and accuracy.

This experiment does not have any comparison with other research, hence, the result using a different dataset will become divergent.

Before the dataset got the augmented, the performance score for both models giving the poor results, due to lack of data. After do the augmentation with inserting around 40000 additional datasets, both models giving a good result gaining 70%> for all metrics score, as shown in Table 6 and 7.

The vanilla XLNet model reaches 6 epochs without being penalized by the early stopping, while the XLNet and fastText model reaches 8 epochs. Moreover, in this case, XLNet with fastText give a better result than the vanilla version of XLNet.

In conclusion, the combination of XLNet and fastText embeddings significantly improves gender classification on Twitter data. The proposed method outperforms the baseline XLNet model, especially after data augmentation, suggesting that incorporating additional data and utilizing advanced embedding techniques can enhance model performance.

Table 6: Results comparison for baseline XLNet & XLNet-fasttext in gender classification by twitter feed

| Model | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| XLNet | %70.49 | %77.02 | %59.81 | %67.48 |
| XLNet-FT | **%71.42** | **%77.04** | **%60.94** | **%68.05** |

Table 7: Results comparison for baseline XLNet & XLNet-fasttext in gender classification by user description

| Model | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| XLNet | %70.54 | **%77.01** | %59.86 | %67.4 |
| XLNet-FT | **%72.42** | %75.1 | **%63.24** | **%68.6** |

Our proposed method, XLNet-fastText, shows improved performance compared to the baseline XLNet model and several other SOTA methods discussed in the related works. Specifically, our model achieves an accuracy of 71.4%, precision of 77%, recall of 60.9%, and F1-score of 68% for gender classification based on Twitter feeds. For user account descriptions, the scores are 72.4% for accuracy, 75.1% for precision, 63.24% for recall, and 68% for F1-score.

In comparison, Movahedi Nia et al. (2022) achieved an accuracy of 79.94% using BERT for text classification and 85.52% when combining text and images. Similarly, Staykovski T. (2019) reported an accuracy of 85.96% by combining TF-IDF and Doc2Vec. Our results, while lower, highlight the potential of integrating fastText with XLNet to enhance context understanding, especially in purely text-based classifications.

The integration of FastText embeddings with XLNet provides several advantages, including better handling of word semantics and context in textual data. This is evident from the improved recall and F1-scores compared to the baseline XLNet model. However, our method underperforms compared to multi-modal approaches like that of Movahedi Nia et al. (2022), indicating that incorporating additional data types, such as images, could further enhance classification performance.

One of the notable strengths of our model is its relatively high precision, which suggests that the model is effective at correctly identifying instances of gender. However, the recall score indicates room for improvement in correctly identifying all relevant instances, hinting at potential biases in the training data or model limitations in generalizing across diverse text inputs.

The observed performance differences can be attributed to several factors. Firstly, the multi-modal approaches leverage additional contextual information from images, which is not utilized in our current model. This additional data helps in better disambiguating gender cues that might not be evident from text alone.

Secondly, the choice of datasets and their preprocessing steps play a crucial role. Our dataset, derived from Kaggle and preprocessed to remove non-relevant labels, might lack the diversity and volume needed to train the model effectively. Data augmentation techniques were applied to mitigate this, but further enhancements in data quality and diversity could lead to better model performance.

Our proposed XLNet-fastText model introduces a novel approach by combining the transformer-based XLNet with fastText embeddings to improve gender classification on Twitter. This combination leverages the strengths of both models: the contextual understanding of XLNet and the semantic richness of fastText. Our approach demonstrates a viable pathway to enhance text classification models without relying on multi-modal data.

# 6 Conclusion

In short, the experiment shows that the XLNet give a good performance on the data with a small number of classes. Besides, XLNet-fastText obtain better performance and fastText helpful for this experiment.

Several things can be concluded after doing this research, namely:

- XLNet-fastText overcomes the gender classification on Twitter
- Both proposed models obtain with very close score
- The quality of data really affecting the training performance for the models

Several improvements in features engineering and parameter tuning could potentially be advancing the research. The first is increase the experiment with another transformers model such as BERT and XL model and other XLNet models to compare this research with the same dataset. The last is use more and better quality of the dataset to get a better performance.

# References

[1] Delić, D. (2022). Are women at more risk of online scams? The latest 2024 statistics. Retrieved from https://proprivacy.com/blog/women-and-online-scams-latest-statistics-2022

[2] Susandra, A. (2022). Erayani Pelaku Penipuan nikah Sesama Jenis dilaporkan ke Polresta Jambi : Okezone Video. Retrieved from https://video.okezone.com/play/2022/06/30/1/149948/erayani-pelaku-penipuan-nikah-sesama-jenis-dilaporkan-ke-polresta-jambi

[3] Yang, L., Li, Y., Wang, J., & Sherratt, R. S. (2020). Sentiment analysis for e-commerce product reviews in Chinese based on sentiment lexicon and deep learning. *IEEE Access*, 8, 23522–23530. https://doi.org/10.1109/access.2020.2969854

[4] Bazzaz Abkenar, S., Haghi Kashani, M., Akbari, M., & Mahdipour, E. (2023). Learning textual features for Twitter spam detection: A systematic literature review. *Expert Systems with Applications*, 228, 120366. https://doi.org/10.1016/j.eswa.2023.120366

[5] Adhikari, A., Ram, A., Tang, R., Hamilton, W. L., & Lin, J. (2020). Exploring the limits of simple learners in knowledge distillation for document classification with DocBERT. *Proceedings of the 5th Workshop on Representation Learning for NLP*. https://doi.org/10.18653/v1/2020.repl4nlp-1.10

[6] Joshi, S., & Abdelfattah, E. (2021). Multi-class text classification using machine learning models for online drug reviews. *2021 IEEE World AI IoT Congress (AIIoT)*. https://doi.org/10.1109/aiiot52608.2021.9454250

[7] Suleymanov, U., Kiani Kalejahi, B., Amrahov, E., & Badirkhanli, R. (2020). Text classification for Azerbaijani language using machine learning. *Computer Systems Science and Engineering*, 35(6), 467–475. https://doi.org/10.32604/csse.2020.35.467

[8] Garcia-Mendez, S., Fernandez-Gavilanes, M., Juncal-Martinez, J., Gonzalez-Castano, F. J., & Seara, O. B. (2020). Identifying banking transaction descriptions via support vector machine short-text classification based on a specialized labelled corpus. *IEEE Access*, 8, 61642–61655. https://doi.org/10.1109/access.2020.2983584

[9] Zhong, B., Xing, X., Love, P., Wang, X., & Luo, H. (2019). Convolutional neural network: Deep learning-based classification of building quality problems. *Advanced Engineering Informatics*, 40, 46–57. https://doi.org/10.1016/j.aei.2019.02.009

[10] Wani, A., Joshi, I., Khandve, S., Wagh, V., & Joshi, R. (2021). Evaluating deep learning approaches for COVID-19 fake news detection. *Combating Online Hostile Posts in Regional Languages during Emergency Situation*, 153–163. https://doi.org/10.1007/978-3-030-73696-5_15

[11] Gupta, A., Chugh, D., Anjum, & Katarya, R. (2022). Automated news summarization using transformers. *SpringerLink*. https://doi.org/10.1007/978-981-16-9012-9_21

[12] Anwar, M. T., Permana, A. K., Ambarwati, L., & Agustin, D. (2021). Analyzing public opinion based on emotion labeling using transformers. *2021 2nd International Conference on Innovative and Creative Information Technology (ICITech)*. https://doi.org/10.1109/icitech50181.2021.9590110

[13] Kumar, D., Kumar, N., & Mishra, S. (2021). NLP@NISER: Classification of COVID-19 tweets containing symptoms. *Proceedings of the Sixth Social Media Mining for Health (#SMM4H) Workshop and Shared Task*. https://doi.org/10.18653/v1/2021.smm4h-1.19

[14] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). ALBERT: A lite BERT for self-supervised learning of language representations. *arXiv.org*. https://doi.org/10.48550/arXiv.1909.11942

[15] Yao, T., Zhai, Z., & Gao, B. (2020). Text classification model based on fastText. *IEEE Conference Publication*. https://doi.org/10.1109/ICAIIS49377.2020.9194939

[16] Nia, Z. M., Ahmadi, A., Mellado, B., Wu, J., Orbinski, J., Agary, A., & Kong, J. D. (2022). Twitter-based gender recognition using transformers. *arXiv.org*. Retrieved from https://arxiv.org/abs/2205.06801

[17] Vashisth, P., & Meehan, K. (2020). Gender classification using Twitter text data. *2020 31st Irish Signals and Systems Conference (ISSC)*. https://doi.org/10.1109/issc49989.2020.9180161

[18] Puertas, E., Ureña-López, L. A., Pomares-Quimbaya, A., Alvarado-Valencia, J. A., Plaza-del-Arco, F. M., & Moreno-Sandoval, L. G. (2019). Bots and gender profiling on Twitter using sociolinguistic features. *PAN at CLEF 2019*. Retrieved from https://www.researchgate.net/publication/335611800

[19] Staykovski, T. (2019). Stacked bots and gender prediction from Twitter feeds. *CEUR-WS.org*. Retrieved from https://ceur-ws.org/Vol-2380/paper_197.pdf

[20] Alroobaea, R., Aldahass, A., Alhomidi, S., Alafif, S., Hamed, R., Mulla, R., & Alotaibi, B. (2020). A decision support system for detecting age and gender from Twitter feeds based on comparative experiments. *International Journal of Advanced Computer Science and Applications*, 11(12). https://doi.org/10.14569/ijacsa.2020.0111245

[21] Saeed, U., & Shirazi, F. (2019). Bots and gender classification on Twitter. *Notebook for PAN at CLEF 2019*. Retrieved from https://pan.webis.de/downloads/publications/papers/saeed_2019.pdf

[22] Ouni, S., Fkih, F., & Omri, M. N. (2022). Bots and gender detection on Twitter using stylistic features. *Advances in Computational Collective Intelligence*, 650–660. https://doi.org/10.1007/978-3-031-16210-7_53

[23] Soldevilla, I., & Flores, N. (2021). Natural language processing through BERT for

identifying gender-based violence messages on social media. *2021 IEEE International Conference on Information Communication and Software Engineering (ICICSE)*. https://doi.org/10.1109/icicse52190.2021.9404127

[24] Hashempour, R., Amorim, R., Villavicencio, A., & Plank, B. (2019). A deep learning approach to language-independent gender prediction on Twitter. *ACL Anthology*. https://aclanthology.org/W19-3630

[25] Eight, F. (2016). Twitter user gender classification. *Kaggle Dataset*. Retrieved from https://www.kaggle.com/datasets/crowdflower/twitter-user-gender-classification

[26] Benkaddour, M. K. (2021). CNN-based features extraction for age estimation and gender classification. *Informatica*, 45(5), 759–770. https://doi.org/10.31449/inf.v45i5.3262

[27] Mathew, L., & Bindu, V. R. (2022). Efficient transformer-based sentiment classification models. *Informatica*, 46(8), 119–128. https://doi.org/10.31449/inf.v46i8.4332