**FastText Model for Offensive Language Identification**

The FastText model is initially trained using the public training datasets called OLID and SOLID. OLID and SOLID are public datasets containing annotated tweet text data. Then the trained model is subsequently evaluated on the SOLID test dataset. This trained model can be used to determine whether the text data contains offensive language, is targeted offensive language, and the target of the offensive language. This feature can help platform users identify whether there is offensive language in the text data they upload, what type of offensive language, and what suspicious offensive language words exist.

The training, evaluation and inference depend on Python 3.7 and the packages including pandas, numpy, fasttext, re, nltk, sklearn, tqdm, typing.

1. **Datasets**

We use OLID and SOLID datasets for model training and testing. OLID (Offensive Language Identification Dataset) is a dataset to classify offensive language on social media [1]. It contains 14,100 English tweets that have been manually labeled using a three-level taxonomy:

* Offensive Language Detection
* Categorization of Offensive Language
* Offensive Language Target Identification

Although OLID has been widely used, it's relatively small in size and might be biased toward offensive language as the data was collected using specific offensive keywords. Also, because of the hierarchical nature of its taxonomy, the number of instances decreases at more specific levels, making it difficult to train robust deep learning models.

SOLID (Semi-Supervised Offensive Language Identification Dataset) is a larger dataset, containing over 9 million English tweets [2]. The creation process for SOLID was semi-supervised, using OLID as a seed dataset to avoid time-consuming manual annotation. SOLID, being larger and collected in a more principled manner, can significantly improve the performance of offensive language identification models.

1. **FastText Model**

FastText [3] is an extension to the word-based model that uses subword representations, making it particularly effective in handling the noisy structure of tweets. FastText is a subword model that has demonstrated robust performance across various tasks without extensive hyperparameter tuning. It utilizes a shallow neural model for text classification, similar to the continuous bag-of-words model. Unlike the continuous bag-of-words model that predicts a word based on its surrounding words, FastText predicts the target label based on the words in the sample. Figure 1 shows the model architecture of FastText.

FastText's ability to break words down into smaller chunks (subwords) allows it to better understand and represent words that may have been misspelled or written in a non-standard way, a common occurrence in social media language. This is particularly useful in the context of offensive language detection, where people might deliberately misspell words or use non-standard forms to evade detection.

One major reason to choose FastText is its fast training speed. For large-scale text data, FastText can complete training in a relatively short time. Additionally, FastText uses character-level n-grams features, enabling it to understand words not present in the language model, making it highly suitable for handling social media text full of innovative usages and typos.

However, FastText can't capture relationships between sequences of words (i.e., it doesn't take into account the order of words). For complex semantic texts, FastText might not provide deep and precise understanding.

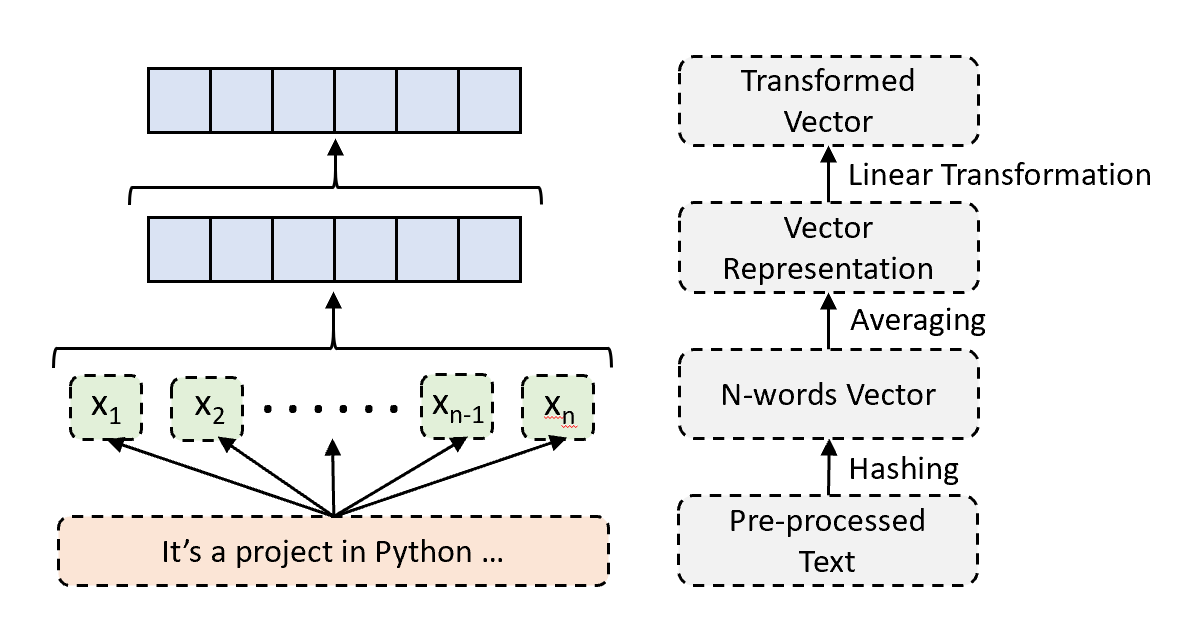


Figure 1. FastText Model Architecture

1. **Training**

In the SOLID paper, FastText was trained with bigrams and a learning rate of 0.01 for Levels A and B of the OLID/SOLID taxonomy. For Level C, it is trained with trigrams and a learning rate of 0.09. All tasks use a window size of five and a hierarchical softmax loss. In the implementation, We trained FastText with the default values of hyperparameters as they can achieve similar prediction performance.

The training sets from OLID and SOLID were combined (though it’s optional to use either one of them for training). This combined dataset was then randomly shuffled before being used for model training. Enhancements in the performance for Levels B and C were achieved by implementing an upsampling strategy for the classes that were underrepresented. For each class, we sampled 'K' instances where 'K' represents the count of instances belonging to the most frequently occurring class. This method helped to balance the representation of different classes in the data.

1. **Evaluation**

We evaluate the performance of each model by comparing the predicted labels with the actual labels for the SOLID test set. The evaluation results for the FastText model at different tasks are as follows:

* Task A: The model demonstrates high performance with an accuracy of 0.92. It has a precision, recall, and F1-score of 0.99, 0.84, and 0.91 for the 'NOT' class, and 0.86, 0.99, and 0.92 for the 'OFF' class respectively.
* Task B: The model's performance drops compared to Task A, with an overall accuracy of 0.54. For the 'UNT' class, the precision, recall, and F1-score are 0.53, 0.92, and 0.68 respectively. For the 'TIN' class, these metrics are 0.62, 0.15, and 0.24, indicating a lower recall and F1-score.
* Task C: The model's accuracy is 0.60. For the 'IND' class, the precision, recall, and F1-score are 0.48, 0.61, and 0.54 respectively. For the 'GRP' class, these metrics are 0.85, 0.63, and 0.73, showing a relatively good performance. For the 'OTH' class, these metrics are 0.13, 0.30, and 0.19, which are significantly lower, indicating the model's poor performance on this class.

In summary, the FastText model performs well on Task A but demonstrates lower performance on Tasks B and C, particularly struggling with the 'OTH' class in Task C.

References

[1] Zampieri, Marcos, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. "Predicting the Type and Target of Offensive Posts in Social Media." In *Proceedings of NAACL-HLT*, pp. 1415-1420. 2019.

[2] Rosenthal, Sara, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. "SOLID: A Large-Scale Semi-Supervised Dataset for Offensive Language Identification." In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 915-928. 2021.

[3] Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov. "Enriching Word Vectors With Subword Information." *Transactions of the Association for Computational Linguistics* 5 (2017): 135-146.