Improving Reddit Post Classification via Deslanging

W266 Final Project
04.05.22
Nathan Chiu, Kevin Fu, Allison Schlissel
Github Repository: https://github.com/apschlissel/w266-final-project

Abstract:

Our research seeks to improve Reddit category (“Subreddit”) classification accuracy. We used Naive Bayes as a baseline model, and ran multiple experiments using various algorithms and dataset configurations that “deslang” the self-posts on various models, including BERT, T5, and RNN. We “deslanged” the dataset of the Reddit self-posts using a regular expression (“regex”) to replace the slang with its non-slang definition from a modified slang to non-slang synonym translation list from the website https://slangit.com. We compared performance between the four models on the original dataset and the deslanged dataset. Our results suggest that our deslanged models did not outperform the original text models in most cases, and in the few cases in which the deslanged models outperformed against the original text models, the improvements were modest. Further research could involve improving the fidelity of translations when deslanging texts.

Introduction:

Online forums such as Reddit have grown in significance, especially amidst the social distancing of the pandemic for discussing ideas, coordinating events, and sharing information. The posts on this forum are organized by subject into categories called "subreddits" (such as news, movies, pets). Keeping subreddits on-topic elevates the user experience by providing relevant information and posts to interact with. Moreover, better post classification has implications for a wide variety of fields ranging from better understanding informal language to online moderation to better organizing social media content.

Currently, most forums such as Reddit use moderation bots to govern channels or subreddits in this case. These bots usually use the number of posts and number of upvotes a user has received from other users before approving a post in the subreddit. However, this may not be the best way of moderating content since there is a time lag. A better way to evaluate content relevance might be to scan through the body text real-time to determine, without additional input, if the post belongs in that category. Much of the language in these forums contain slang, hence our research paper’s exploration of whether or not slang would have an impact on classification performance.

Background:

Slang in NLP is an emerging field with data scientists developing slang corpuses[1], slang generators[2], and slang sentiment analysis[3]. Slang can confound language models since they are not typically in a formal dictionary. In terms of other scholars' approach to textual understanding and classification involving slang, most employ different dictionaries and bag-of-words approaches to achieve their NLP objectives such as sentiment analysis to text classification[4]. In text classification surveys, scholars have identified misspellings and slang as significant barriers to textual classification[5].

At the same time, NLP models have become more and more powerful. Scholars have used deep learning models to understand texts with slang.[6] Recurrent neural networks (RNNs) have sequential architecture that enables retention of information, but can falter with longer sentences and larger contexts. Building on RNNs, transformer models such as BERT have proved faster and more accurate than RNNs via pre-training on corpuses. The emergence of transformer models has also sparked development of newer transfer learning models.In particular, transfer learning models like T5 offer a promising avenue for better text classification. The key difference between T5 and BERT is that T5 reframes tasks as a one to one text task, whereas BERT only outputs "either a class label or a span of the input"[7]. The three models have various tradeoffs and the paper compares their performance on regular social media texts and deslanged texts.

Approach/Methods:

We created an experiment in which we ran four different models (Naive Bayes, BERT, T5, RNN) on two different datasets: one with the original text and another with slang words replaced by their non-slang definitions, resulting in deslanged text.

Our hypothesis is twofold:

1) BERT, T5, and RNN will perform better (higher F1 score) than Naive Bayes baseline on subreddit classification on the same text datasets. We use the F1 score instead of the accuracy since the former accounts for both false positives and negatives.
2) All models will perform better on the "deslanged" texts i.e. the original Reddit posts with slang replaced with their definitions.

---

[1] Wilson, S., Magdy, W., McGillivray, B., Garimella, K., & Tyson, G. (2020). Urban dictionary embeddings for slang NLP applications. LREC 2020 - 12th International Conference on Language Resources and Evaluation, Conference Proceedings, 4764-4773. https://doi.org/10.17863/CAM.56914

[2] Sun, Zhewei, Richard Zemel, and Yang Xu. "A Computational Framework for Slang Generation." *Transactions of the Association for Computational Linguistics* 9 (2021): 462-478.

[3] Märtens, Marcus, et al. "Toxicity detection in multiplayer online games." *2015 International Workshop on Network and Systems Support for Games (NetGames)*. IEEE, 2015.

[4] Wu, L., Morstatter, F. & Liu, H. SlangSD: building, expanding and using a sentiment dictionary of slang words for short-text sentiment classification. *Lang Resources & Evaluation* 52, 839–852 (2018). https://doi.org/10.1007/s10579-018-9416-0

[5] Kowsari, Kamran, et al. "Text classification algorithms: A survey." *Information* 10.4 (2019): 150.

[6] Parvathi, R. "Deep Learning for Social Media Text Analytics." *Advanced Deep Learning Applications in Big Data Analytics*. IGI Global, 2021. 68-91.

[7] Google AI

Data
To collect the data for the project, we scraped posts and associated subreddit labels from Reddit.com.

Because Reddit has such a large variety of posts and we had limited data storage and compute power, we had to limit the amount of subreddits and ensure the subreddits we chose spanned a wide range of subjects. We focused our analysis on 3 groups of subreddits:

1. 5 handpicked subreddits that we thought had the most amount of slang within the posts: 'wallstreetbets', 'teenagers', 'GenZ', 'copypasta', 'unpopularopinion'. These will hereinafter be referred to as "**5 Handpicked Subreddits**".
2. 5 subreddits that were very similar to each other: 'gaming', 'PS4', 'pokemon', 'xboxone', and 'leagueoflegends'. These will hereinafter be referred to as "**5 Similar Subreddits**".
3. 5 subreddits that were randomly selected by our program out of the top 500 subreddits in terms of number of subscribers: 'Bitcoin', 'memes', 'travel', 'philosophy', 'stocks'. These will hereinafter be referred to as "**5 Random Subreddits**".

For each of the 3 groups above, we pulled 500, 1000, and 5000 posts from each of the 5 subreddits (2,500, 5,000, and 25,000 posts total). We did not pull more than 5000 posts due to modeling time[8] and memory[9] constraints. After pulling these dataset configurations, we saved them to a series of comma separated value files (.csv) to ensure uniform datasets among the team members.

We searched extensively for data that offered current slang and a direct non-slang synonym. We found such a compilation from Slangit, and then modified the compilation as a group to remove irrelevant or uncommon slang terms. Upon finalizing the Slangit data, we created a deslanged dataset using a regular expression that replaces the slang word with its non-slang synonym.

Baseline
Originally, we used the BERT model to classify 100 reddit posts in each of the five subreddits in the slang-heavy **5 Handpicked Subreddits**. This baseline model generated an accuracy of roughly 60% and we were motivated to improve model performance through various model and dataset changes.

Upon feedback, we pivoted to use a simpler model, the Multinomial Naive Bayes, as our baseline. The accuracy measures remained very similar between this more classical ML model and our initial BERT model; both averaged around 60-65%.

Modeling
After building the baseline models, we added additional dataset configurations and models to further improve model performance.

We added two additional datasets, the **5 Similar Subreddits** and the **5 Random Subreddits**, to test for model robustness and to prevent overfitting to those five slang-heavy subreddits. Overall,

[8] running 1 epoch of the Transformers models took well over 45 minutes to complete
[9] running 1 epoch of the T5 model on large datasets was ram-intensive, crashing instances

using the **5 Random Subreddits** improved our model performance because the subreddits were comparatively different from each other. As expected, the models had a difficult time classifying the **5 Similar Subreddits** due to the significant overlap in category (all gaming-related).

BERT: We used an uncased BERT model since a slang heavy data set will be prone to many random capitalizations. Therefore, lowercasing everything was a good idea. A max length of 256 tokens proved to be more than sufficient in terms of the F1 score. Due to machine constraints, datasets of 500 & 1000 posts ran for 3 epochs, and the datasets of 5000 posts ran for 1 epoch.

T5: The T5 (Text-to-Text Transfer Transformer) model builds upon transfer learning models such as BERT and reframes text classification as a text-to-text problem. Since subreddit names and reddit posts are both text, we thought that this classification would be an appropriate T5 application. We used only one epoch to increase runtime speed and had a batch size of 128 (though various iterations of batch size did not change model performance, nor run time).

RNN: We incorporated two LSTM layers for better performance for the shallower datasets i.e. 500 posts per subreddit. Then we added a dense layer with relu activation followed by a softmax pooling layer. We experimented with batch size too, but observed marginal benefits..

Results & Discussion:

Final Results Table: F1 Scores on Test Set
80/10/10 train/val/test

Subreddit	n per subreddit	Original Text Naive Bayes	Deslanged Naive Bayes	Original Text BERT	Deslanged BERT	Original Text T5	Deslanged T5	Original Text RNN	Deslanged RNN
5 Handpicked subreddits ['wallstreetbets', 'teenagers', 'GenZ', 'copypasta', 'unpopularopinion']	5000	0.684	0.654	0.833	0.826	0.772	0.731	0.719	0.716
5 Handpicked subreddits	1000	0.473	0.436	0.706	0.692	0.644	0.684	0.493	0.592
5 Handpicked subreddits	500	0.532	0.519	0.741	0.747	0.616	0.624	0.522	0.516
5 Similar subreddits ['gaming' 'PS4' 'pokemon' 'xboxone' 'leagueoflegends']	5000	0.753	0.734	0.793	0.793	0.713	0.666	0.685	0.66
5 Similar subreddits	1000	0.6	0.591	0.684	0.685	0.612	0.576	0.583	0.602
5 Similar subreddits	500	0.735	0.711	0.772	0.739	0.688	0.68	0.594	0.612
5 Random subreddits ['Bitcoin', 'memes', 'travel', 'philosophy', 'stocks']	5000	0.722	0.712	0.842	0.84	0.784	0.816	0.728	0.732
5 Random subreddits	1000	0.574	0.558	0.776	0.765	0.75	0.736	0.655	0.651
5 Random subreddits	500	0.653	0.623	0.96	0.952	0.872	0.916	0.788	0.7
Average Score across All Models:		0.636	0.615	0.79	0.782	0.717	0.714	0.641	0.642

Figure 1: F1 scores across the four models and two datasets.

We achieved our highest F1-scores using the BERT model. This was no surprise given that it was pre-trained on 3B+ words for specifically this task and BERT is able to learn the word order bidirectionally, whereas Naive Bayes only takes a bag of words approach (no direction)[10]. Especially within the **5 Similar Subreddits** category, it is surprisingly impressive that the BERT model is able to still maintain a ~80% F1 score even with the similarities and overlapping of those subreddits.

Lessons from first iterations

[10]Winastwan, Ruben. “Text Classification with Bert in Pytorch.” Medium, Towards Data Science, 10 Nov. 2021, https://towardsdatascience.com/text-classification-with-bert-in-pytorch-887965e5820f.

- Deslanging texts during our first iterations resulted in some nonsensical or uncommon translations. We mitigated this by manually cleaning some texts of symbols that messed up our models such as apostrophes in contractions. In instances of slang with more than one definition, we preserved the one that was more commonly used.
- Naive Bayes: Naive Bayes was performing extremely well after the first iteration because we were vectorizing based on the Dataframe's index values as opposed to the word text's vectorization values.
- BERT: When a reddit post is deleted by the author, what remains is the words "[removed]" or "[deleted]". This phenomenon ends up being misclassified by BERT. Removal of these essentially null posts added around a 5-6% gain in F1 score.
- RNN: First iterations resulted in the same results or NaN loss values because of data shallowness and incompatible data. We fixed these issues by one hot encoding and adding the dense softmax layer in the sequential model.

Patterns observed in results

- Across the board, deslanging the texts did not help model performance.
- BERT systematically outperformed both T5 and RNN. This aligns with our hypothesis since BERT is more dynamic especially with larger contexts than RNN and T5 isn't designed for text classifications.
- Posts that were not in English or posts that were too short (one word or a short phrase containing 5 or less words) were commonly misclassified. For example, "Philosophy" was the most error-prone category, and most misclassifications were in the category of "Memes." The posts that were incorrectly misclassified as "Memes" had an average word length of 30.2, whereas the average post within the "Philosophy" category had an average word length of 105.5. This makes sense because most posts within the Meme category have a much shorter word length (average word length of 22.4). *See appendix.*

Error Analysis:

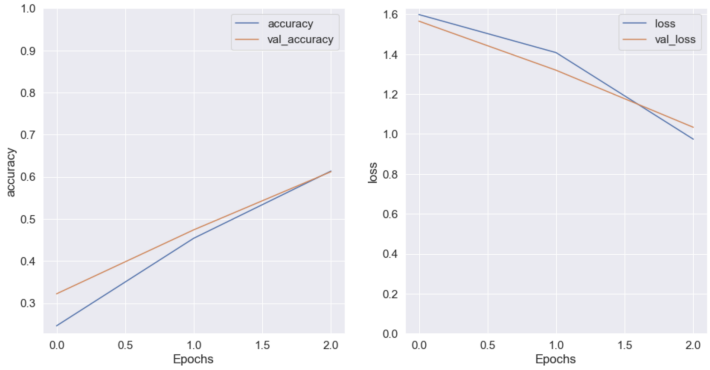


Figure 2: RNN Accuracy and loss graphs over 2 epochs for n = 500 for 5 Similar Subreddits.

For the 5 Similar Subreddits, which expected the RNN model to have a more difficult time classifying, training loss declined over epochs, but we encountered GPU issues with larger datasets such as the n = 1,000 and n = 5,000 datasets that forced us to use 1 epoch in the hyperparameters.

Given greater computation power, we would use a higher number of epochs to improve the performance of our models.

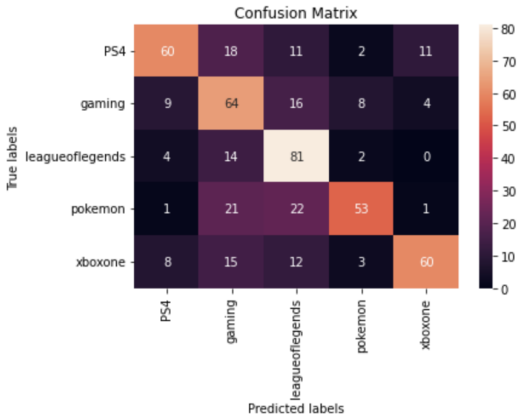


Figure 3: T5 model (deslanged dataset, similar subreddits, n=1000) Confusion Matrix

Figure 3 shows the results of the worst performing T5 model. Upon further examination of results, the most commonly misclassified subreddit was pokemon, often with league of legends or gaming. We found that the pokemon subreddit often had reddit posts that seemed irrelevant to pokemon, perhaps because of the waning popularity of Pokemon GO. For examples, *see appendix*.

Conclusions:

Despite significantly improving over our baseline Naive Bayes model, we were unable to reject the null hypothesis. Deslanging the Reddit posts actually made all the models across the board, with exception of a few instances in the RNN, underperform the models that used the original text. The Transformer models likely used the slang words as unknown tokens <unk> to correctly classify the models into the correct subreddit (category). Since the BERT models are likely pretrained on regular English language models, it is unlikely that these slang-based tokens appeared in its original vocabulary. This is contrary to our original hypothesis: if all text were translated to “normal” English words, the model will be able to decipher it better and perform better classifications.

Secondly, translating slang as a task on its own proved to be more difficult than originally thought. Like regular English, there may be multiple definitions of a single slang word (eg. “zzz” could mean either “I am sleeping” or “this is boring” depending on the context). There may be conflicts between an actual english word vs. the slang word (eg. “we” the pronoun and “we” as an abbreviation for “whatever”). There may be multiple slang words with the same token (eg. “lol” as slang for “laugh out loud” and “lol” as an abbreviation for popular online game League of Legends).

Therefore, the challenges above contributed to our Deslanged models underperforming the Original Text models even though the Original Text models significantly outperformed the Naive Bayes baseline model.

References:

Chavez, Gustavo. “Implementing a Naive Bayes Classifier for Text Categorization in Five Steps.” *Medium*, Towards Data Science, 6 Mar. 2019, https://towardsdatascience.com/implementing-a-naive-bayes-classifier-for-text-categorization-in-five-steps-f9192cdd54c3#:~:text=Naive%20Bayes%20is%20a%20learning,%2C%20%E2%80%9CPromotions%E2%80%9D%2C%20etc.

Wilson, S., Magdy, W., McGillivray, B., Garimella, K., & Tyson, G. (2020). Urban dictionary embeddings for slang NLP applications. LREC 2020 - 12th International Conference on Language Resources and Evaluation, Conference Proceedings, 4764-4773. https://doi.org/10.17863/CAM.56914

Sun, Zhewei, Richard Zemel, and Yang Xu. "A Computational Framework for Slang Generation." *Transactions of the Association for Computational Linguistics* 9 (2021): 462-478.

Märtens, Marcus, et al. "Toxicity detection in multiplayer online games." *2015 International Workshop on Network and Systems Support for Games (NetGames)*. IEEE, 2015.

Wu, L., Morstatter, F. & Liu, H. SlangSD: building, expanding and using a sentiment dictionary of slang words for short-text sentiment classification. *Lang Resources & Evaluation* 52, 839–852 (2018). https://doi.org/10.1007/s10579-018-9416-0

Kowsari, Kamran, et al. "Text classification algorithms: A survey." *Information* 10.4 (2019): 150.

Parvathi, R. "Deep Learning for Social Media Text Analytics." *Advanced Deep Learning Applications in Big Data Analytics*. IGI Global, 2021. 68-91.

Appendix:

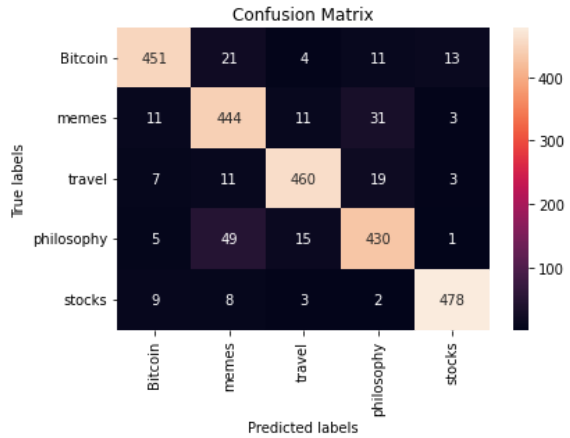


Figure 3: Confusion matrix, Random Subreddits

- Spammy-sounding posts were classified as bitcoin posts - which is pretty accurate in today's world.
- Incorrect "philosophy" classifications within memes were within reason; they sounded somewhat philosophical to a human reader.
- After deslanging, short posts continue to be misclassified as memes, at a 50% higher rate than before. Average post word length within philosophy category = 124, longer than before. Misclassification to memes had an average word length of 35.

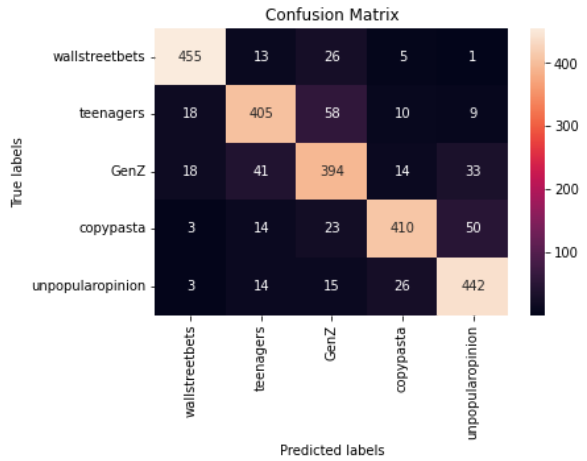
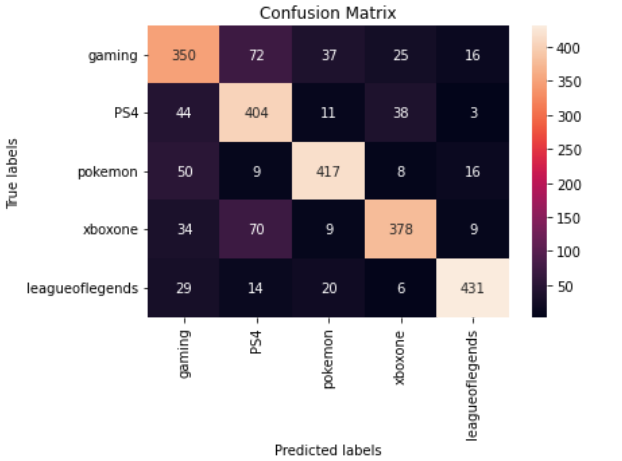


Figure 4: Confusion matrix, Handpicked Subreddits

- GenZ and Teenagers are the most confused, and appropriately so.
- Very long posts are classified as unpopularopinion since most of the time, those posts are generally longer than the other ones. The average length of wallstreetbets, GenZ, and teenagers were around 280-360 characters long, whereas the average length of unpopularopinion was double that, around 606 characters long. The longest was actually copypasta, which had 1363 characters long.
- Copypasta actually does very well in spite of its long length, likely because most posts are either repetitive (hence the name), or plain gibberish.
- Many of the incorrect copypastas actually originated from wallstreetbets or sounded like a teen wrote them. Therefore, this score is likely on par with what a human reader would have scored.



	precision	recall	f1-score	support
0	0.69	0.70	0.70	500
1	0.71	0.81	0.76	500
2	0.84	0.83	0.84	500
3	0.83	0.76	0.79	500
4	0.91	0.86	0.88	500
accuracy			0.79	2500
macro avg	0.80	0.79	0.79	2500
weighted avg	0.80	0.79	0.79	2500

Figure 5 & 6: Confusion matrix, Classification Report: Similar Subreddits

- As expected, the similar subreddits did the worst out of the 3 subreddit groups.
- Gaming was the lowest performer since it encapsulated all four of the other subreddit groups. However, it was interesting that it got mixed up with the PS4 posts 50% more often than the Xbox One posts.
- The second worst performer was the Xbox One posts which often got mixed up with Ps4 posts. The surprising thing was the vice versa was not true: PS4 posts got mixed up with Xbox One posts only half as much.
- Upon review of the posts, most of the incorrect categorizations technically are correct due to how similar the posts are.

Results from Baseline Report:

```
Epoch 1
Training loss: 1.4655039935884342
Validation loss: 1.3714709329605101
F1 Score (Weighted): 0.3039062778193213

Epoch 2
Training loss: 1.2078518615642064
Validation loss: 1.1382132411003112
F1 Score (Weighted): 0.5469613526570049

Epoch 3
Training loss: 1.0220267149344298
Validation loss: 1.0786930060386657
F1 Score (Weighted): 0.5939885476841998

Epoch 4
Training loss: 0.9038504210995956
Validation loss: 1.0680918085575104
F1 Score (Weighted): 0.5595399698340875

Epoch 5
Training loss: 0.8501881980140444
Validation loss: 1.0331812393665314
F1 Score (Weighted): 0.599581939799331
```

```
Class: wallstreetbets
Accuracy: 6/15

Class: teenagers
Accuracy: 1/15

Class: copypasta
Accuracy: 12/15

Class: GenZ
Accuracy: 2/15

Class: unpopularopinion
Accuracy: 4/15
```

Exhibit 1 Baseline: BERT model on original dataset, n = 500, 100 each in five subreddits. Accuracy was determined on a test set of n = 15 as a result of the 85/15 train/test split.

The baseline BERT model has an accuracy of 60% on the untranslated dataset. The model did fairly well on the “copypasta” subreddit. When the model was first run, unsurprisingly, some of the posts had the same text and that artificially inflated the accuracy. When we corrected for duplicate posts, the model still did well with the subreddit. The model does very poorly on the teenagers, GenZ, unpopularopinion subreddits.

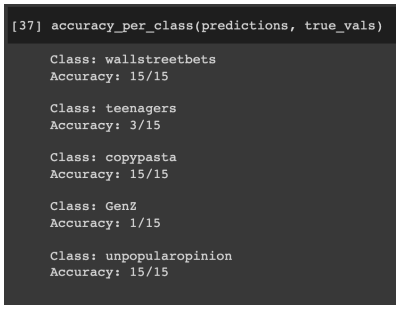


Exhibit 2 Baseline: BERT model run on deslanged dataset

As seen in the results above, BERT did a lot better on classifying the Reddit posts when we found and replaced the slang words with their definitions. Accuracy drastically dropped for Gen Z, but improved across most of the five subreddit labels.

Pokemon subreddit misclassification examples:

- Cleaning services, maids nyc
- Well i can’t deal with it snapchat, well I can’t deal with it
- What I first thought when the title dropped