Slang Identification and Definition using BERT

Eric Clark
CSCI 573
Western Washington University
clarke21@wwu.edu

Alexis Ewan
CSCI 404
Western Washington University
alexisewan0228@gmail.com

Abstract

Creating a better understanding of slang terms is beneficial to many people, and it is a relatively unperformed work. Pre-existing works detect slang terms in a sentence, or slang terms in a text, but they are not effective at identifying which word is the slang term. Furthermore, existing works do not offer a solution to defining slang terms. We developed a BERT language model to identify slang terms in a sentence and define those terms for the user. To do this, we used Urban Dictionary, an online crowd-sourced slang database, to catalog slang terms and their definitions. We enlisted BIO tagging to label each word in the sample sentences as slang or not slang. We then trained the Hugging Face BertForTokenClassification pre-trained model based on bert-base-uncased to classify tokens in an input sentence. The resulting model produced a Matthew's Correlation Coefficient score of 0.870 and a weighted F-score of 0.9997 when classifying slang terms in reserved testing samples. Finally, we queried an Urban Dictionary dataset for definitions of the identified slang terms and presented those definitions to the user.

1 Introduction

Slang is an informal vocabulary used to distinguish cultural groups, hence it is an ever-evolving subject in natural language. The issue with slang words is that they can be ambiguous, challenging to understand, and dynamic. Our solution is to use a language model to identify the slang terms in a sentence, and then supply the user with the definition of those terms. To achieve this, we trained a BERT model with a mix of Google News and Urban Dictionary sample sentences. The model labels sentences to identify the slang terms, and then we use Urban Dictionary definitions to define those terms.

The results of this project can benefit many different people. For example, people with ADHD or ASD often struggle to understand slang terms because they are an abstraction of social contexts. Also, it can be difficult for people who are learning English as a second language to understand slang terms. Finally, slang is very indicative of human culture, and studying slang and its evolution can be very beneficial towards sociology and cultural anthropology.

2 Existing Work

Existing work for this topic includes a paper entitled *Slang detection and identification* (Pei et al., 2019) and a paper entitled *Detection of Slang Words in e-Data using semi-Supervised Learning* (Pal and Saha, 2017). Each of these papers take a very different approach to detecting slang in sentences, and their work gives us good insight on the applications of language models in detecting slang terms.

Slang detection and identification (Pei et al., 2019) uses a binary classification task to identify whether a sentence contains a slang term or not. They do this by labeling each term in the sentence as either slang or not slang, and then determine the classification of a sentence based on the presence of slang labels, or lack thereof. They implemented this using a BiLSTM-MLP model and a BIO labeling convention. Additionally, they chose to include part of speech (POS) tagging, bigrams, unigrams, and pointwise mutual information as features in their model. In the end, their models exhibited an average F-score of 0.5111 when identifying the specific slang word in a sentence, and an average F-score of 0.8675 when detecting sentences that contain slang terms.

Detection of Slang Words in e-Data using semi-Supervised Learning (Pal and Saha, 2017) contrar-

ily takes an algorithmic approach to slang detection for the sake of monitoring government communication systems. The algorithm is intended to filter abusive or suspicious language amongst internal government communications. This algorithm includes first detecting jargon words outright using a jargon database, then using a sounds-alike jargon database to filter words that are intentional or accidental misspellings of the jargon words. Then the algorithm reports those words and notifies the user to remove them from the text. Finally, the algorithm parses the text again for potentially unrecorded jargon words (suspicious terms) and suspicious words are determined to be slang or not by a human. If the suspicious word is a slang term, it is added to the slang database, and otherwise it is used to enrich the learning set. This learning set is used to better the algorithm by fine-tuning its jargon-detection process.

Our takeaways from these works include a deeper understanding of past and present practices in slang detection. In our work, we adapted the use of BIO tagging to label our samples with a set of 3 tags. We considered using POS tagging to aid our model, but determined it was outside the scope of this project. We found that using slang term and sentence databases as the basis of a model was standard, so we found a slang database on Kaggle to use in our model. What is explicitly missing from these existing works is the presence of slang definitions. Being able to detect slang is important, but being able to define the terms themselves for a better understanding of the text is a critical task that is overlooked by existing works.

3 Approach & Experiments

In order to develop our model, we had to preprocess our data first. We began by using an Urban Dictionary (hereafter referred to as UD) vocabulary set and an UD API to query data from the UD website. This provided us the information to create a dictionary with UD vocabulary words as the keys and example sentences for those words as the values. After we obtained the positive sample sentences, we applied beginning-inside-outside (BIO) tags to each sentence. In doing so, we identified non-slang terms (outside), individual slang terms (beginning), and slang phrases (beginning & inside) for each sentence. We also applied BIO tags to a set of Google News sentences to form our negative samples. After we had the appropriate labels for the set of UD sentences and the Google News sentences, we combined both sets to create the FullSamples dataset, which contains a total of 89,659 sample sentences.

Once we obtained the full set of labeled samples, we split the data 80/20 to form our training and testing sets. We used the BERT tokenizer to encode each sentence in the training set, and then we used those encodings to develop our TensorDataset. Then we further divided our training set into a 90/10 training-validation split. We loaded the bert-base-uncased pretrained model and then trained it with our testing samples to fine-tune. Training the model took less than two-hours, and after it was done we decided to export the model so it could be easily loaded in the future without rerunning the training code.

Our final tasks involved testing, evaluating, and utilizing our model. We used our model on the 20% of testing data we set aside earlier to predict labels for each sentence. We then used the Matthews Correlation Coefficient function from sklearn to evaluate our model. On average, our model produces an MCC of 0.86. Finally, we developed a function that applies the model to a given sentence, then uses the labels of that sentence to identify the slang terms. Once we identified the term, we used the original UD vocab set to define the term.

4 Experiments

4.1 Dataset

UD vocab set - dataset from Kaggle

Slang-positive samples - UD slang definition example sentences. (before pre-processing, see examples.txtin supplementary materials) - 14,878 sentences

Slang-negative samples - We used the Google News dataset provided in assignment 2 (see train.txt and test.txt is supplementary). - 74,781 sentences

4.2 Pre-Processing

The UD sample sentences are not well-formed, as they are user-generated. They contained conversation snippets between multiple people as example usages, as well as odd formatting with colons, blocks of hyphens, multiple newlines, and other non-useful tokens. In order for our model to have a better chance of understanding the data, we cleaned it up some.

For preprocessing, we converted everything to lowercase, and stripped out non alphanumeric characters, except hyphens for reasons explained later. We then broke up the sentences into words, and used the word the sentence is an example for to detect slang within the sentence, for BIO labeling. At this stage, it became apparent that we needed to keep hyphens in our dataset, as 376 of the sample slang phrases contained hyphens in them, and removing them from the sentences was creating issues matching the slang phrase to its usage in the sentence. The same pre-processing was applied to the slang-negative samples, where BIO labeling was created of exclusively Outside tokens, on the assumption that news stories are not going to contain slang.

See datasetProcessing.ipynb for the preprocessing and dataset importing code.

4.3 Experiment

We used the BERT tokenizer from HuggingFace to tokenize all of our input sentences before they were input to the BERT model. Some of the dataset sentences (219) were over 512 tokens long, which is the max length BERT supports. These sentences were removed from the dataset before it was split. Our full dataset was split randomly into model training and test data, on an 80-20 split. The model training data was further split into training and dev sets, on a 90-10 ratio. This gave us a final dataset split of:

Total	89,659 sentences
Training	64,554 sentences
Dev	7,173 sentences
Test	17,392 sentences

We used the HuggingFace BertForTokenClassification pre-trained model based on bert-base-uncased, to predict the BIO label for each token in the input sentence. The model was trained on the training data over 4 epochs, with a batch size of 32. The AdamW optimizer was used, with a learning rate of 2e-5 and an epsilon value of 1e-8.

The model was then evaluated using the test data split.

We used Matthews correlation coefficient, as well as weighted precision, recall, and fscore to evaluate our model. The precision, recall and fscore were weighted, as we have a large class imbalance towards 'O', as most of our inputs were not slang words.

5 Results & Discussion

5.1 Quantitative

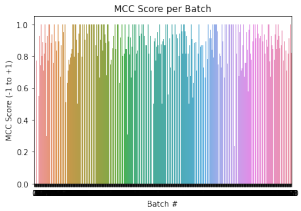

Figure 1: Graph of MCC per Batch

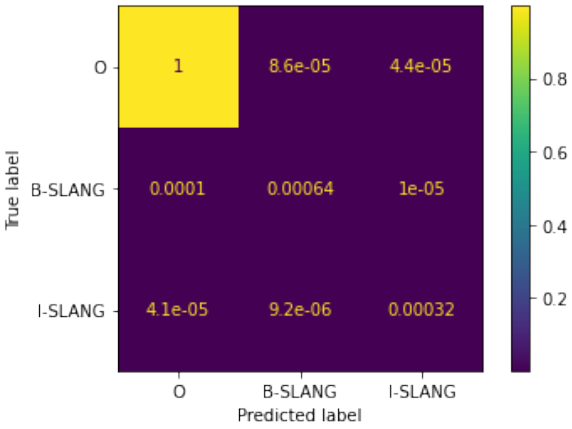

Figure 2: Confusion matrix of all test data

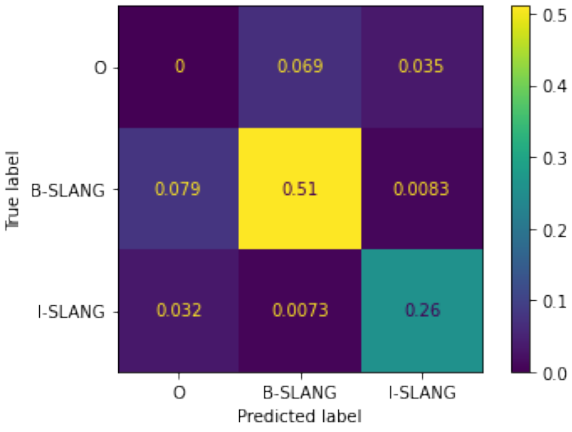

Figure 3: Confusion matrix with correct O label predictions removed

Figure 3 allows us to get better insights about the data, as the information is not shrunk to uselessness by the large number of correct O labels (3,349,066/3,353,284) as in 2. We think this number is so large due in part to the sentences being padded by the bertTokenizer before being input to the model.

From this diagram, we can see that of the remaining data, 77% was correctly identified as slang, 11.3% was slang that was identified as not slang by the model, 8.7% was the beginning of a slang phrase that was incorrectly identified, and 3.9% was inside slang that was incorrectly identified.

Result metrics calculated using sklearn.metrics functions:

MCC: 0.870

Weighted Precision: 0.9997

Weighted Recall: 0.9997

Weighted Fscore: 0.9997

5.2 Qualitative

When looking at the results returned from the predictions, some of them work, like

```
if you you re walking down the
    street and you re approaching
    an attractive group of ladies
    pop the collar and you re in
    there
1 slang words found
The slang word is: "pop the
    collar"
The definiton of pop the collar
    is: defined in moving images..
     www.youtube.com/watch?v=
    tNgzKlKPhJY


you know epf
1 slang words found
The slang word is: "epf"
The definiton of epf is: a
    combination of ehh, yep/yup,
    and pff. ;; a sign of
    indifference/sarcasm. ;; used
    best when replying to a
    question or statement that you
     don't really care about. ;;
    for use while typing (text, or
     online)
```

And

```
a noobologist will observe the
    behavior of noobs to futher
    the study of noobology
1 slang words found
The slang word is: "noobologist"
The definiton of noobologist is:
    Someone who studies noobs to
    futher the study of noobology.
because of his beef tooth mike
    never order pineapple or
    eggplant toppings but instead
    always ordered sausage or
    pepperoni pizza
1 slang words found
The slang word is: "beef tooth"
The definition of beef tooth is:
    Similar to a [sweet tooth],
    but for meat.
```

However some of the predictions seem to be using the wrong definition of the word, as in

```
that disgusting stewed spinach
    made me gerg
1 slang words found
The slang word is: "gerg"
The definition of gerg is: Verb:
    To finish 4th in a multi-table
     poker tournament.
```

Where the true definition (based on which gerg definition this came from) is "A mini-vomit you catch in the back of your throat or mouth." This issue is due to the large number of definitions for some words on Urban Dictionary, especially anything close to a name.

Also, some predictions seemed off in terms of which word in the sentence was being selected as slang:

```
i heard that jenny is a bi-trans
    said sara wait isn t your mom
    bi-trans since u said she
    switched sexes and loves both
    said tara don t judge said
    sara
2 slang words found
The slang word is: "bi-trans"
The definition of bi-trans is: a
    person who has switched sexes
    and loves both sexes


The slang word is: "mom"
The definition of mom is: The
    woman who wants you to get
    something that's 5 feet away
    from her

today is the seventh day of
    kwanzaa imani what s your name
     oh my name is imani
```

```
2 slang words found
We do not have a definition for "
    kwanzaa imani". Sorry!
```

Where the first bi-trans is found correctly, but the second is predicted at the index of mom. This is probably due to issues with encoding/decoding by the BERT word piece tokenizer not matching up exactly with words.

Sometimes the model predicted label would end too early, as in

```
in the pink or in the stink i can
     t decide lets have a greasy
    grundle gamble by the
    grundlefull
1 slang words found
We do not have a definition for "
    greasy grundle". Sorry!
```

Where the actual slang phrase is "greasy grundle gamble", rather than just greasy grundel.

I think a good amount of these issues are not with the model, but with the decoding from the tokenized version of the sentence, and modifying the corresponding token predictions to true-up to the actual sentence input. Currently our system just takes the predictions, which are the classifications for the tokenized input, and then ties them one to one with the words in the input, rather than accounting for anything the word piece tokenizer is doing that isn't just splitting on spaces.

5.3 573 Discussion

Overall, these results are very good, with high scores for all metrics. However, we think the data these metrics are calculated from is tainted due to the predictions being done on the tokenized input, rather than the raw sentences. Due to this, padding tokens are included, which the model is going to be very good at identifying as Outside, which artificially inflates the scores. However, BERT is very powerful at language classification, and our results show that. Improvements to the model will mostly lie in data preprocessing, and postprocessing, as we do not think we would be able to improve on the science and engineering behind BERT very much. These very high accuracy scores are not necessarily reflected in the number of problems with the qualitative results but these issues seem to mostly be pre/post processing issues, rather than core problems with the model we have fine tuned.

5.4 Fairness

A fair AI system would not have different outcomes if used by different groups or classes of people. It would not have biases in its learned representations, and its results would not benefit or malign specific groups. I think our system is not as fair as it could be for a couple reasons. First, our model is based on a pre trained BERT model, so it has all of the inherent biases and fairness issues that the base BERT model. Second, our datasets have some bias. For slang positive samples, they are biased towards groups that have the time and inclination to add definitions or example usage to Urban Dictionary, so our data exhibits selection biases. We are sure that not all slang is captured on Urban Dictionary, and the writing style of the examples is not representative of everyone. For slang negative samples, they are based on news reports, and so carry the selection biases of what is reported on in the news. Finally, the Urban Dictionary definitions we are presenting to the user represent the biases of the community who created them. To improve the fairness of the system, we would like to gather definitions from multiple slang dictionaries to hopefully represent a wider audience. We would also like to fine-tune the model on other slang example sources.

5.5 What we Learned

During the course of this experiment, we learned a lot of technical skills for training a dataset, such as preprocessing the data, training the model, and evaluating that model. However, the bigger lesson here is the complexity of building a model to identify slang terms in sentences. During the course of this experiment, it became evident that developing an effective slang classifier requires a robust dataset to formulate training and testing sets. That is to say that what dataset you use is just as important as how you use it. We spent several days researching datasets and trying to find a precompiled dataset, and we ended up combining a couple of datasets to form our sample set. Creating our own set in this manner also allowed us to configure the dataset to fit our needs instead of vice-versa.

This experiment also provided us an opportunity to experience the amount of computational power needed to train a model. Our dataset was relatively small, with less than 100,000 samples instead of millions of samples, yet it still took hours for our model to train. We chose to export our model af-

ter we trained it the first time, but reloading that model requires several minutes. Because of this extended loading time, it is unfeasible for this model to be used as a browser extension. We believe that our model is better suited to be a web API where the computations can be done remotely or with a virtual machine rather than the user's personal machine.

Finally, we learned that the model we created was feasible and actually successful in its purpose. When we started this project, we were unsure if developing a model to identify slang would be possible due to the complexity of slang terms. Slang terms do not always fit the laws of natural language, and sometimes it can be hard to determine what words are slang terms or not. However, this model was able to do just that, and it ended up being better at it than we expected.

6 Conclusion & Future Work

We were able to train a model that had a large amount of success in detecting which words in a sentence were slang. When evaluated more qualitatively, the model was a little less impressive, but still useful at the task. This success was with a relatively small dataset of around 85,000 sentences, we assume that with a larger dataset, the effectiveness would be increased. We have thought of a number of ways to improve the work, as well as extend it.

- **user input** - We would like to incorporate user feedback in our predictions, in order to further fine tune the model as it is used
- **definition disambiguation** - Adding additional tagging or information to the model output to help us disambiguate between multiple definitions of the same slang term. We want to investigate POS tags and bigram or other contextual information.
- **in-place definition** - We would like to clean up the definitions and support replacing the slang term in place with the definition, or a common language word with the same meaning
- **tokenization handling** - We would like to update the tokenization encode/decode process to better handle expanding and shrinking labels to ensure the labels match up exactly to the sentence, rather than the tokenizer word piece output.
- **evolving updates** - We would like both the model and dictionary to have the capability to update over time to recognize new slang terms. Slang is ever-changing, our work would be more useful if it kept up.

7 References

The following were used as examples, tutorials, and general how-tos for using the model and data processing:

- (Briney, 2022)
- (Kulkarni, 2017)
- (et al, 2019)

(Wolf et al., 2020) is the HuggingFace transformers library whose pre-trained BERT model we utilized for our work.

References

Victor Sanh et al. 2019. Run$_g lue.py$. Hugging Face/-Transformers/Examples from Github. Version 4.17.0.

Seth Briney. 2022. Copy of fine$_t une_b ert.ipynb$. Google Colab.

Rohit Kulkarni. 2017. Urban dictionary words and definitions. Kaggle.

Alok Ranjan Pal and Diganta Saha. 2017. Detection of slang words in e-data using semi-supervised learning. *CoRR*, abs/1702.04241.

Zhengqi Pei, Zhewei Sun, and Yang Xu. 2019. Slang detection and identification. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, page pages 881–889.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

A Supplemental Material

SlangIdentification.ipynb
datasetProcessing.ipynb
Google Drive with saved BERT model and all of the datasets