NLP ALBUMENTATION

ISM6930 TEXT ANALYTICS – PROJECT REPORT

UNIVERSITY of SOUTH FLORIDA

Submitted By
Shailendra Singh and Enock Chemochek

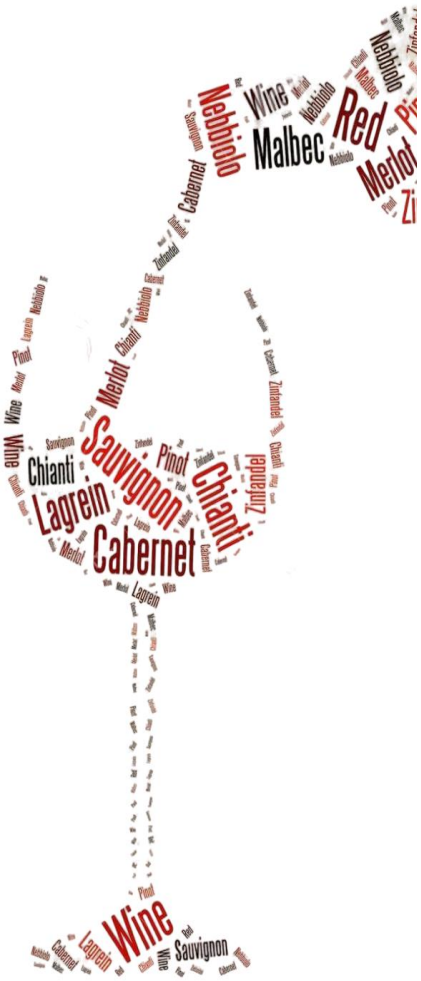


TABLE OF CONTENTS

01. EXECUTIVE SUMMARY

NLP Albumentation is a data augmentation technique inspired from image augmentation techniques that are quite popular in the field of computer vision. Augmentation in case of text data is called albumentation, and it is a field of natural language processing which focusses on producing quality data from a limited amount of available NLP Albumentation is a data augmentation technique inspired from image augmentation techniques that are data by trying to keep the same context or grammatical structure.

To generate new text data from existing data, we used Yelp reviews as our dataset to apply text augmentation techniques. The idea is to add noise to the current data, to generate similar but different data. The idea is to generate noise in the existing data by using supervised augmentation techniques such as back translation, synonym-antonym replacement, etc.

Our goal is to generate similar but more data from existing set of Yelp reviews so that any classification model that is built upon the existing dataset can leverage the availability of the new data to improve its classification metrics such as accuracy, recall, and other metrics. We first build a base model to generate some metrics and then will use the augmented text to feed in the new model and compare the metrics with the original model.

Our hope is to get similar or better metrics. If we get similar metrics, it would also be a success since augmentation does not require drastic change in the original data. The techniques we use try to keep the same semantic meaning of the text by applying minimal changes. The techniques also try to keep the grammatical structure somewhat the same as the original data.

02. PROBLEM DEFINITION & SIGNIFICANCE

In recent years, the rapid development of models for AI technology has their use in various fields of communication, finance, medicine, and so on. However, insufficient data poses a big problem due to the challenges in the data collection process. When the number of samples collected does not meet the needs of model training, the model will be in the state of underfitting. Also, data imbalance in classification task leads to low accuracy and recall rate.

Data augmentation is one technique that can help with such problems by increasing the training samples while also reducing the imbalance of sample count among the different classes (if it is a classification problem). By obtaining more high-quality data, augmentation practices can help the model to improve the model's robustness.

Since the internet is mostly image data, the first ideas of augmentation came from generating more images so that the neural networks work better at classifying them. The techniques used in computer vision includes translating, rotating, compressing, or adjusting the color of RGB channel in the images. However, in the case of text there is a dearth of quality data due to its discrete nature; the same techniques are impossible to apply with text data.

Taking the ideas of augmentation from computer vision, we can augment text data and this field is called NLP albumentation. NLP albumentation is an area which focuses on using augmentation techniques to generate more high-quality data from a given limited amount of data.

03. PRIOR LITERATURE

There has been a lot of research in the field of NLP albumentation. The techniques can be classified in the following categories:

1. Supervised
2. Semi-Supervised
3. Unsupervised
    a. Generating new data
    b. Learning enhancement strategies

Semantic exchange is a technique proposed by Feng et al., 2019 [1] in which the semantics of the text are adjusted while preserving the text polarity as well as its fluency.

Keyword replacements using synonyms and antonyms as described in Liu, Meng, 2020 [2] changes the polarity of sentences by flipping synonyms and antonyms.

Back-translation as described in J Son, 2018 [3] is one of the most widely used enhancement techniques in which a text input is converted sequentially in various languages, and finally converted into the original language of the text. The idea is to add noise due to the different grammatical structures of various languages, thus keeping the same semantic meaning while altering the structure of the input text.

04. DATA SOURCE

We used a subset of Yelp data [4]. This dataset contains user reviews of various businesses and services such as automobile repair, restaurant service, hotel service, and so on. The reason for choosing this dataset is because in text augmentation the ideal length of the input should be medium. If it is too short, then the semantic keyword replacement would not work as there is not much context to work with. If it is too long, then the back translation technique wouldn't work as there is a limit on the number of characters with which the trained multi-lingual models work.

We used only the text column as shown below:

```
print(yelp_df.shape)
yelp_df[['text']].head()
```

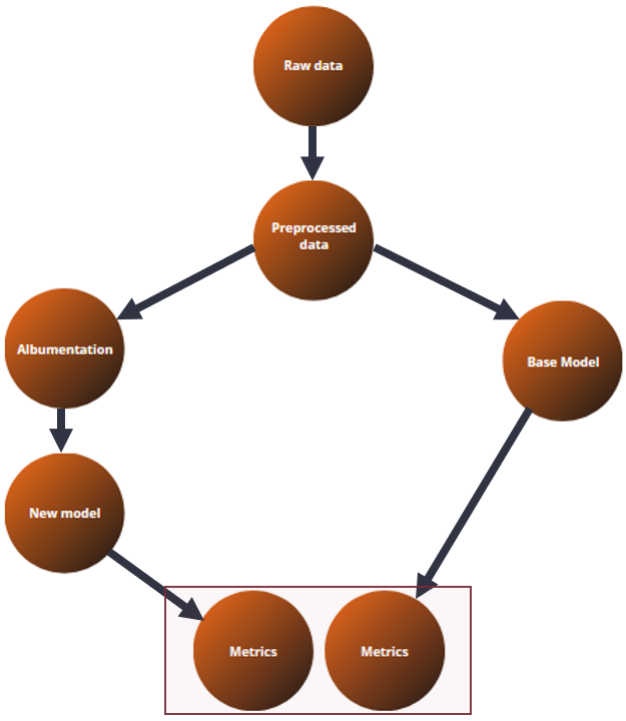
```
(10000, 15)
```

	text
0	My wife took me here on my birthday for breakfast and it was excellent....
1	I have no idea why some people give bad reviews about this place. It go...
2	love the gyro plate. Rice is so good and I also dig their candy selecti...
3	Rosie, Dakota, and I LOVE Chaparral Dog Park!!! It's very convenient an...
4	General Manager Scott Petello is a good egg!!! Not to go into detail, b...

5 rows × 1 columns Open in new tab

05. TEXT ANALYTICS WORKFLOW

Following is the workflow for comparing metrics:



As explained in the sections above, we seek to create synthetic data from existing data to have more data on which we can train models and expect them to perform better by virtue of 'seeing' more examples to learn from. Given that the data we are interested in contains only text which could only be possibly used for unsupervised machine learning algorithms, we sought to give data the label indicative of sentiment – positive, negative or neutral. The text analytics workflow explained below is reflective of this complexity. The problem we are solving requires a two-fold approach; creating text augmentations and measuring success of it. The second part requires training a select model on data before the data augmentation process and taking note of the metrics, which will then be compared with those from a model trained on augmented data.

PREPROCESSING

The preprocessing step was mainly focused on removing any existing null values and making the text all lower case. Stop words were not removed since that would affect the steps that come after this, that is sentiment analysis and the albumentation.

EXPLORATION

The exploration phase included looking at most frequent words, phrases (bigrams and trigrams) as well as some named entity recognition processing. We explored the linguistic structure and constitution of these texts to get a better sense of what parts of speech (POS) were the most frequently used and which ones were less used. This information contributes towards the intimate understanding of the dataset.

SENTIMENT ANALYSIS AND LABELING

The text data were labeled according to sentiment, as a starting point for building a classifier on which to compare the performance of our albumentation solution. In this step the data was classified into 3 – positive, negative and neutral – based on our self-defined formula below.

$$Label = \begin{cases} \text{Negative} & \text{Polarity} < \text{- } 0.2 \\ \text{Neutral} & \text{-}0.2 < \text{Polarity} < +0.2 \\ \textit{Positive} & \text{Polarity} > +0.2 \end{cases}$$

Note: The labels were then appended to the data for downstream processing.

BASE MODELS

Two approaches towards modeling were adopted: supervised learning (using sentiment labels assigned in the earlier steps) and unsupervised learning (can we improve within and between cluster metrics with albumentation?). The architecture of these models was simple since the focus of our study was not on the absolute best performance of these models, but rather comparing the performance given different sets of data (original versus augmented). Clustering was performed at the document level because augmentations was largely focused on the document level.

ALBUMENTATION

The data that is used for building the base models as described above is also taken downstream for the albumentation step, after which new models of the same architecture as the base models are trained on this expanded dataset.

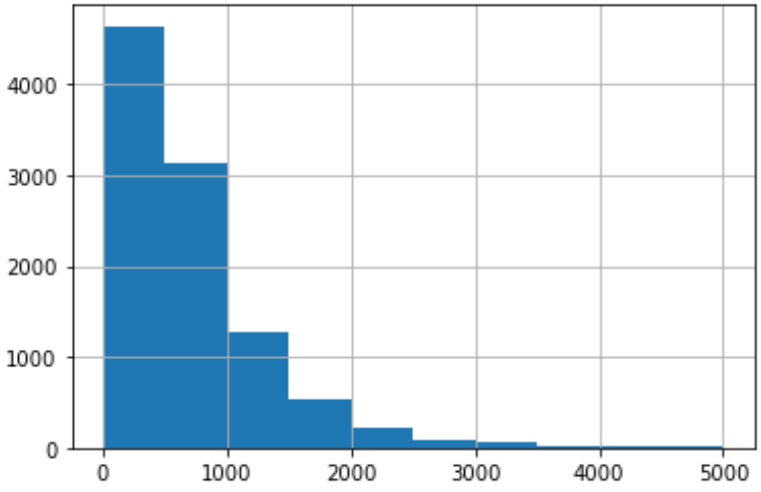
METRICS

The data that is used for building the base models as described above is also taken downstream for the albumentation step, after which new models of the same architecture as the base models are trained on this expanded dataset. Similar metrics are compared between the base and the new models.

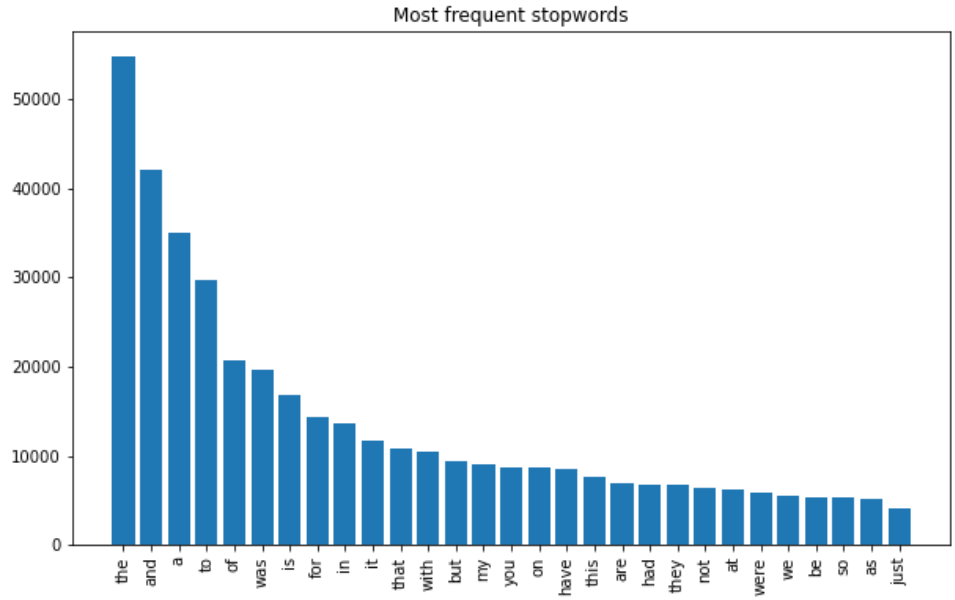
06. EXPLORATORY DATA ANALYSIS

The task of albumentation demands an intimate understanding of the data for it to be a success. We performed exploratory analyses and came up with the following insights:

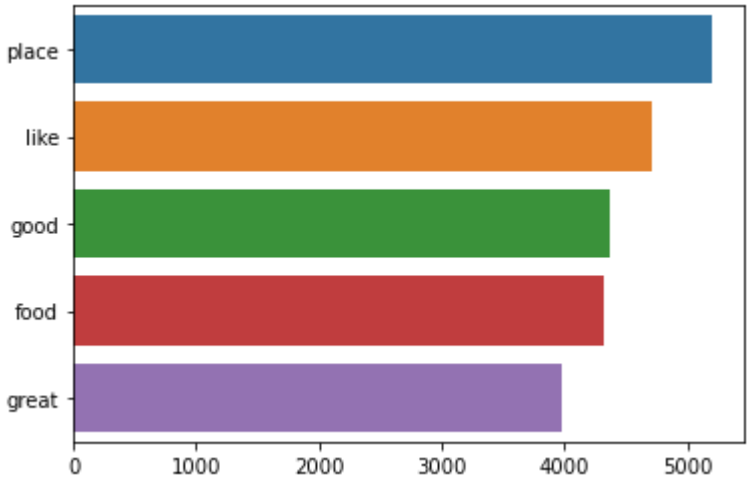
The distribution of length of the reviews (number of characters in the review) shows most reviews have less than 1000 characters before the distribution tails out. This was to the expected behavior of the data.



Stopwords form a big chunk of the sentences in the reviews used as data, and sentiment labeling will also be performed with the text still having all these words, just as will be the albumentation. From the distribution of these words, we notice that 'the' is the most used stopword followed by 'and' and a.



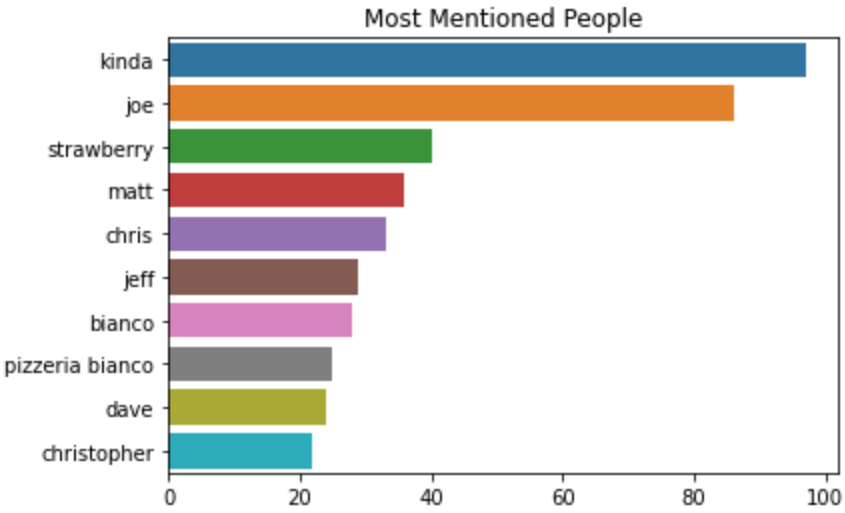
The distribution of non-stopwords is summarized by the plot below:



Users seem to be talking more about the places (or business establishments), their sentiments and the food. This gives a sense of the data, largely being about food outlets or restaurants – which we also expected. The like, great and good we see in the chart is indicative of general positive sentiments of the reviews.

NAMED-ENTITY RECOGNITION ANALYSIS

We performed an analysis of the named entities in the dataset which showed that the most frequent entities were cardinal followed by date & time, and persons respectively. A more critical look into the distribution of persons mentioned in the reviews yielded the following visualization.



Pizzeria Bianco is a restaurant in Arizona that was founded by Chris Bianco, so it makes sense that the 3 words appear the most here. The analysis also showed that among the most mentioned places in the dataset was Scottsdale

07. AUGMENTATION TECHNIQUES AND RESULTS

The following shows three augmentation techniques that were used to generate new data.

Original Review:

```
In 6  1  review = "Fast and nice service. Clean facilities. Super cute shop with all the pastels & cat themed
           art! Cookies and cream was more chocolate, and strawberry was a stronger candy/artificial flavor. It
           melted pretty fast so we had to eat quickly."
```

Note: The highlights on images are done manually. Our code does **not** highlight the changed text.

A. SEMANTIC KEYWORD REPLACEMENT

This technique replaces nouns or noun-phrases using the Universal Sentence Encoder (USE) as it has transformer model using deep averaging network (Cer et al., 2018) [5].

1. The first step is to generate the parts of speech tags for all the words. Using the Stanford Constituency Parser [6], we were able to fetch the tags for nouns and noun-phrases.
2. For each noun or noun-phrase, we use the Hugging Face's pre-trained glove vectors [7] to generate a potential replacement entity for each word.
3. For every potential replacement entity (RE), we used the USE model to find the best target word in the original text. The USE model uses the context of the word to find the potential targets. We sequentially perform this step until all the REs are exhausted and that generates the first augmented sentence.

```
In 7  augmented_review = NLP_AUG.get_augmented_sentence(review, num_iter=10)

In 8  print(augmented_review)

      Fast and nice service. paintings. Super cute shop with all the pastels & rabbit themed phone!
      candy and earthy was more chocolate, and cherry was a stronger candy/phoneificial flavor. It
      melted pretty fast so we had to eat quickly.
```

B. SYNONYM & ANTONYM REPLACEMENT

This technique replaces Verbs, Adjectives, and Adverbs in the input text by using WordNet from NLTK corpus.

1. The input sentence is processed to get Verbs, ADJ, and ADV parts of speech tags using Stanford Constituency Parser[6]; the list will be our potential replacement entity.
2. Using the WordNet model, for each potential replacement entity (RE), we use the word's lemma to generate the synonyms list.
3. For each REs synonym list, we pick an item randomly and replace it in the input sentence.
4. The input sentence generates its synonym-augmented counterpart.
5. Similarly steps 1 to 3 are performed for antonym list of a replacement entity. This again generates an antonym-augmented counterpart of the input sentence.

Note: Both A's and B's implementation can be found in entity replacement module in our git. Please refer to the Appendix section (A).

Augmentation with Synonyms

```
In 9  syn_augmented = NLP_AUG.augment_with_synonyms(augmented_review)

In 10 print(syn_augmented)

      Fast and Nice service. paintings. Super cunning shop with all the pastels & rabbit theme phone!
      candy and earthy was more chocolate, and cherry was a strong candy/phoneificial flavor. It
      mellow_out pretty fasting thence we bear to run_through quick.
```

Augmentation with Antonyms

```
In 11 ant_augmented = NLP_AUG.augment_with_antonyms(augmented_review)

In 12 print(ant_augmented)

      Fast and nasty service. paintings. Super cute shop with all the pastels & rabbit themed phone!
      candy and earthy was fewer chocolate, and cherry was a impotent candy/phoneificial flavor. It
      unmelted unreasonably slow so we lack to eat slowly.
```

C. BACK-TRANSLATION

The idea of back-translation is to generate semantically similar, but different text. We used Hugging Face's Helsinki Pre-Trained Multilingual Models [8]. We used three languages English, French and German.

1. First convert the text from English to French
2. Second, convert the text from step 1 to German
3. Finally, convert the text from step 2 back to English

Note: The code can be found in the back translation module in our git. Please refer to the Appendix section (A).

Note: The steps 1-3 are implemented within a single function, so the output is directly in English. The following steps are shown for demonstration purposes only.

English to French

```
In 15
# English to French
en_fr_review = back_translation_module._get_translation(syn_augmented, back_translation_module
  .TRANSLATION_EN_FR)
print(en_fr_review)
```

```
Service rapide et agréable. peintures. Super magasin de cunning avec tous les pastels & lapin
thème téléphone! bonbons et terreux était plus chocolat, et la cerise était une forte saveur de
bonbons / phonétique. Il mellow_out assez jeûner de là nous portons à courir rapidement.
```

French to German

```
In 16
# French to German
fr_de_review = back_translation_module._get_translation(en_fr_review, back_translation_module
  .TRANSLATION_FR_DE)
print(fr_de_review)
```

```
Schneller und angenehmer Service. Gemälde. Super Cunning-Shop mit allen Pastell & Hase Thema
Telefon! Süßigkeiten und erdeös war mehr Schokolade, und Kirsche war ein starker Geschmack von
Süßigkeiten / Phonetik. Es mellow_out genug Fasten von dort tragen wir schnell laufen.
```

German to English

```
In 17
# German to English
de_en_review = back_translation_module._get_translation(fr_de_review, back_translation_module
  .TRANSLATION_DE_EN)
print(de_en_review)
```

```
Faster and more pleasant service. Paintings. Super cunning shop with all pastel & rabbit theme
phone! Candy and earthy was more chocolate, and cherry was a strong taste of candy / phonetics.
It mellow_out enough fasting from there we wear fast running.
```

Original (before back-translation; for comparison)

```
In 9
syn_augmented = NLP_AUG.augment_with_synonyms(augmented_review)
```

```
In 10
print(syn_augmented)
```

```
Fast and Nice service. paintings. Super cunning shop with all the pastels & rabbit theme phone!
candy and earthy was more chocolate, and cherry was a strong candy/phoneificial flavor. It
mellow_out pretty fasting thence we bear to run_through quick.
```

08. ACTIONABLE RECOMMENDATIONS

XGBOOST

Input Type	Metrics	Base Model	Model w/ Albumentation
TF-IDF	**F1 Score**	0.5422	0.7389
	Accuracy	0.7328	0.8485
N-Grams	**F1 Score**	0.3835	0.4794
	Accuracy	0.6090	0.7175

KNN

Input Type	Metrics	Base Model	Model w/ Albumentation
TF-IDF	**F1 Score**	0.4009	0.6361
	Accuracy	0.7445	0.7605
N-Grams	**F1 Score**	0.3525	0.5737
	Accuracy	0.6080	0.7330

The above metrics show that the albumentation augmented data performed better than the one with the original dataset. The hope was to get better performing models, shown by better metrics. Using augmented data, the F1 Scores as well as overall accuracy show significant improvement, which is desired.

The results imply that we can add our augmented data to our current data, and that can improve the problem of quality data scarcity in a specific domain; since the data was augmented and not imported from another source. This approach of problem solving proves useful especially in the case of class imbalance in a classification task with text data. Numerical data has different established methods to create synthetic examples to deal with class imbalance such as the Synthetic Minority Oversampling Technic (SMOTE), we argue the aggregation of these steps is the equivalent of SMOTE.

FUTURE WORK

The augmentation techniques we used have tried to retain a similar structure of the input text. In the future, we would like to explore more advanced techniques that are used in popular NLP transformer models like GPT-3, etc., and possibly improve upon them. We would also like to introduce hyperparameters to our code so that the user can control how much variation they need in the augmented data as well as what type of variation they require.

09. REFERENCES

[1]. Steven Y. Feng, Aaron W. Li, and Jesse Hoey. 2019. Keep calm and switch on! Preserving sentiment and fluency in semantic text exchange. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2701–2711, Hong Kong, China. Association for Computational Linguistics.

[2]. P. Liu, X. Wang, C. Xiang and W. Meng, "A Survey of Text Data Augmentation," 2020 International Conference on Computer Communication and Network Security (CCNS), 2020, pp. 191-195, doi: 10.1109/CCNS50731.2020.00049.

[3]. Son, J. (2018). Back translation as a documentation tool. The International Journal of Translation and Interpreting Research, 10(2), 89–100. https://search.informit.org/doi/10.3316/informit.864953916346703

[4]. "Yelp Dataset" Yelp, 27 Oct. 2022, https://www.yelp.com/dataset. Accessed 27 Oct. 2022.

[5]. Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

[6]. "Constituency Parsing - CoreNLP." https://stanfordnlp.github.io/CoreNLP/parse.html, Accessed 27 Oct. 2022.

[7]. “Fse/Glove-Wiki-Gigaword-300 · Hugging Face.” Huggingface.co, huggingface.co/fse/glove-wiki-gigaword-300. Accessed 27 Oct. 2022.

[8]. “Helsinki-NLP (Language Technology Research Group at the University of Helsinki).” Huggingface.co, huggingface.co/Helsinki-NLP. Accessed 27 Oct. 2022.

10. APPENDIX

A. NLP Albumentation Techniques: https://github.com/Shailendra-Singh/project-nlp-albumentation

Since Albumentation tasks were very computationally intensive, we used the existing Yelp dataset and added augmented columns using this code. The updated dataset was used in the following code for model comparison.

To see the Data Processing and code usage demonstration: the jupyter notebook is the starting point (AugmentYelpReviews.ipynb in above repo).

B. Model Comparison and Metrics: https://github.com/echemochek/nlp-albumentation-with-yelp-reviews

This code attempts to use the augmented text in various columns (Semantic Augmented, Synonym Replaced, Antonym Replaced, and Back-Translated text).