# Analyzing Tokenizers for Neutralizing Subjective Bias in Language

*Michelle Julia Ng, mchjl@stanford.edu | Aiden Yew Woei Low, aidenlow@stanford.edu*

Stanford
Computer Science

## Overview

Today, the internet often acts as a catalyst for "fake" or "biased" news. From politics to simple news, we are highly susceptible to biased opinions, which in turn will dictate how we act. Therefore, it is incredibly important that news on the internet be as objective as possible. The purpose of this project is to build a tool that can neutralize subjective linguistic bias in English sentences to enable a more fair lens with which we can see the world.

In Neural Machine Translation, Byte-Pair Encoding has been the de-facto standard tokenization strategy in multilingual translation for its added advantage when handling rare words [4]. For this monolingual bias neutralization task, we further present the neutralization performance between two different tokenization strategies:

(1) splitting on whitespace and punctuation word tokenization
(2) Byte-Pair Encoding subword tokenization [4]

## Background

**Subjective Bias**

We focus on neutralizing two forms of subjective bias as outlined by Recasens et al. 2013. [3]. **Framing Bias,** using subjective words or phrases linked with a particular point of view (like using words like best or deepest or using pilfered from instead of based on), and **Epistemological Bias:** linguistic bias that subtly (often via presupposition) focus on the believability of a proposition.

**Example Tokenization:**

**Input sentence**

"he is antagonizing over work"

**Splitting on whitespace and punctiation [OpenNMT's Default Strategy]**

"he" "is" "antagonizing" "over" "work"

**Byte-Pair Encoding (BPE) subword tokenization [SentencePiece]**

"_he" "_is" "_antagon" "izing" "_over" "_work"

BPE is a compression algorithm adapted to enable open-vocabulary NMT model translation by encoding unknown words as sequences of subword units.

## Data

We utilized the Wikipedia Neutrality Corpus consisting of aligned sentences pre and post-neutralization by English Wikipedia editors[2] . Following the work of past researchers [2, 3] we train and evaluate our system on a subset of WNC where the editor changed (through modification or addition) or deleted a single word in the source text.

We use the same dataset partitions utilized by Pryzant et al (2019). for cross-comparison. This dataset consists of 53,803 training pairs (about a quarter of the WNC), 700 development and 1,000 test pairs. [2].

**Examples of pre and post neutralization sentences:**

**Word Deletion**

**Pre:** *she plays a large role in the amazing legends of dune trilogy, written by brian herbert and kevin j. anderson.*

**Post:** *she plays a large role in the legends of dune trilogy, written by brian herbert and kevin j. anderson.*

**Word Modification**

**Pre:** *There are 75 immigrant settlers*

**Post:** *There are 75 immigrant residents*

## Implementation

We implemented a non-linear Neural Machine Translation (NMT) using OpenNMT, a generic library written in PyTorch. We implemented a 2-layer Long Short-Term Memory (LSTM) model optimized using Adam with an additive attention mechanism. An additive attention mechanism is used based on previous experimental basis showing its systematic advantage over other attention mechanisms [1].

An Adam optimizer is used as it implements empirical learning rate decay strategies which systematically yields better performance [1]. An additive attention mechanism is used based on previous experimental basis showing its systematic advantage over other mechanisms [1].

LSTM cells enable our model to process entire word sequences (sentences), optimal for making predictions at the sentence level as its ability to retain information in the long term accounts for sentence-level linguistic features and structures without the vanishing gradient problem faced by RNNs.

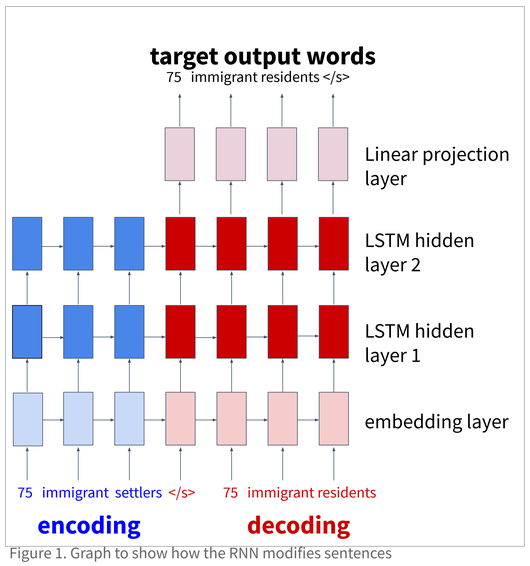


Figure 1. Graph to show how the RNN modifies sentences

**Hyperparameters:**

- **Batch Size:** 16
- **Learning Rate:** 0.005
- **Optimizer:** Adam
- **LSTM Layers:** 2
- **Dropout Probability:** 0.3
- **Vector Length:** 512
- **LSTM Layers:** 2
- **GPU:** Nvidia P100

## Results

We ran three experiments in total in 6 hours of GPU runtime.

| Vocabulary Size | Tokenizer Strategy |
|---|---|
| 3200 | OpenNMT Whitespace Tokenizer |
| 6400 | OpenNMT Whitespace Tokenizer |
| 3200 | BPE Tokenizer |

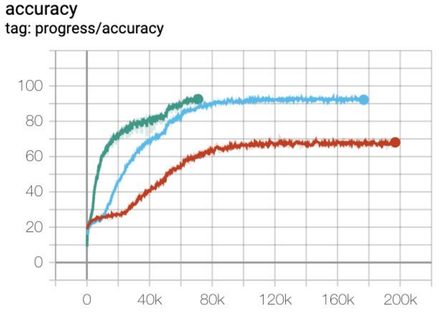


Figure 2. Training Accuracy

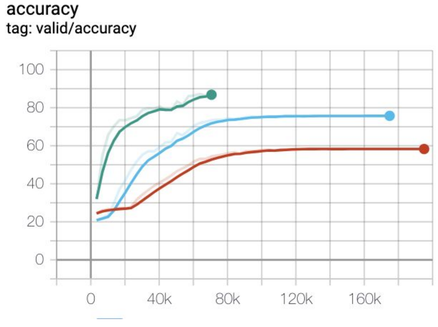


Figure 3. Validation Accuracy

## Evaluation

We evaluate the success of our model using Bilingual Evaluation Understudy (BLEU) scores, perplexity and accuracy. **BLEU score** is a string matching score that measures the fluency of a generated sentence from the reference. **Perplexity** is the measure of how uncertain the model's prediction is. (The lower the better). **Accuracy** is the fraction of predictions our model got right.

| Tokenizer | # Vocab Words | # Epochs | Accuracy | Pred Perplexity | BLEU Score |
|---|---|---|---|---|---|
| OpenNMT | 64,000 | 20 | 68.09% | 1.16 | 74.26 |
| BPE | 32,000 | 20 | 45.21% | 1.38 | 51.04 |
| OpenNMT | 32,000 | 20 | 13.87% | 2.35 | 23.45 |
| BPE | 32,000 | 40 | 50.42% | 1.31 | 56.83 |
| OpenNMT | 32,000 | 40 | 18.82% | 2.0522 | 28.97 |

**Example Subjective Bias Source:**

mark oaten (born 8 march 1964, watford) is a disgraced liberal democrat politician in the united kingdom, and member of parliament for the winchester constituency.

**Target Neutral Case:**

mark oaten (born 8 march 1964, watford) is a liberal democrat politician in the united kingdom, and member of parliament for the winchester constituency.

**OpenNMT Default Tokenization Neutralization (Accurate):**

mark oaten (born 8 march 1964, watford) is a liberal democrat politician in the united kingdom, and member of parliament for the winchester constituency.

**BPE Tokenization Neutralization (Inaccurate):**

mark greyen (born 8 march 1964, ) is a noted democrat politician in the united kingdom, and member of parliament for the wolfe constituency.

## Discussion

- A major challenge in the project was learning PyTorch, understanding OpenNMT with its limited documentation, understanding tokenizer differences, and the large amount of GPU resources required.
- Training moved fastest towards convergence for the 64000 OpenNMT. We theorize this was due to its larger vocabulary size which provided it more inference capabilities as the BPE dataset has not reached convergence.
- Furthermore, our analysis of the dataset also shows that there are situations where BPE makes false calls, such as removing "watford" above, likely due to its subword segmentation which presents a trade-off of being able to make inferences about very rare words while subjecting possibly less common words to unwanted noise leading to incorrect modifications.

## Future Work

- Compare the performance achieved here to that when using vocabulary and a pre-trained model with BERT.
- Conduct pre-training using other domains and toggling hyperparameters to see if it improves metrics for success.
- Use model on news articles and see if people react differently to them. Conduct user interviews to gauge impact.

## References

[1] Britz, D.; Goldie, A.; Luong, M. and Le, Q.Massive Exploration of Neural Machine Translation Architectures. 2017. arXiv:1703.03906v2.

[2] Pryzant, R.; Martinez, R. D.; Dass, N; Kurohashi, S.; Jurafsky, D. and Yang, D. 2019. Automatically Neutralizing Subjective Bias in Text. arXiv preprint arXiv:1911.09709.

[3] Recasens, M.; Danescu-Niculescu-Mizil, C.; and Jurafsky,
D. 2013. Linguistic models for analyzing and detecting
biased language. In Proceedings of ACL, 1650–1659.

[4] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In ACL.