# Optimizing an Automatic Part-of-Speech Tagger for Poetry Text using Data Augmentation

Hirona Jacqueline Arai ('22)
CS702 Senior Thesis (Spring '22)
Advised by John Foley, Ph.D.

## Abstract

| Text: | A | studious | student | studied | in | the | study. |
|---|---|---|---|---|---|---|---|
| POS Tags: | DET | ADJ | NOUN | VERB | PREP | DET | NOUN |

Part-of-Speech (POS) Tagging remains a crucial building block of natural language processing. The accuracy of this machine learning task, in which text is automatically labelled with a corresponding sequence of parts of speech, determines the success of downstream text analysis processes. High-quality off-the-shelf algorithms and libraries for POS tagging is prevalent (SpaCy, Stanford tagger, NLTK). However, because these algorithms have been trained on newspapers and literary text which feature well-formatted sentences, a limitation is that they perform poorly on a typographically different corpus. Poetry text is one example for which off-the-shelf POS taggers have low accuracy. By utilizing high-accuracy part of speech taggers on existing datasets and applying data augmentation[1] that mimic the syntactic features of poems, we efficiently create training data for this domain of resource-limited text.

In this thesis, we train a POS tagger that will perform better on poetry corpora. Using data augmentation methods, we utilize labelling sequences derived from modified Wikipedia text to simplify the process of creating a domain-specific POS tagger. We improve performance from 84% accuracy to 95% accuracy.

## Methods

1. Label Wikipedia text using an off-the-shelf POS tagger:

| Original Wiki | Vermont | is | a | state | in | the | New | England | ... |
|---|---|---|---|---|---|---|---|---|---|
| Off-the-shelf POS tags | NNP | VBZ | DT | NN | IN | DT | NNP | NNP | ... |

2. Augment dataset by performing poetry mutations:

   a. Fragmentation

| Poetry Wiki | Vermont | is | a | state | . | \n | in | the | New | England | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Poetry tags | NNP | VBZ | DT | NN | . | \n | IN | DT | NNP | NNP | ... |

   b. Reduplication

| Poetry Wiki | Vermont | is | a | state | in | the | New | New | England | ... |
|---|---|---|---|---|---|---|---|---|---|---|
| Poetry tags | NNP | VBZ | DT | NN | IN | DT | NNP | NNP | NNP | ... |

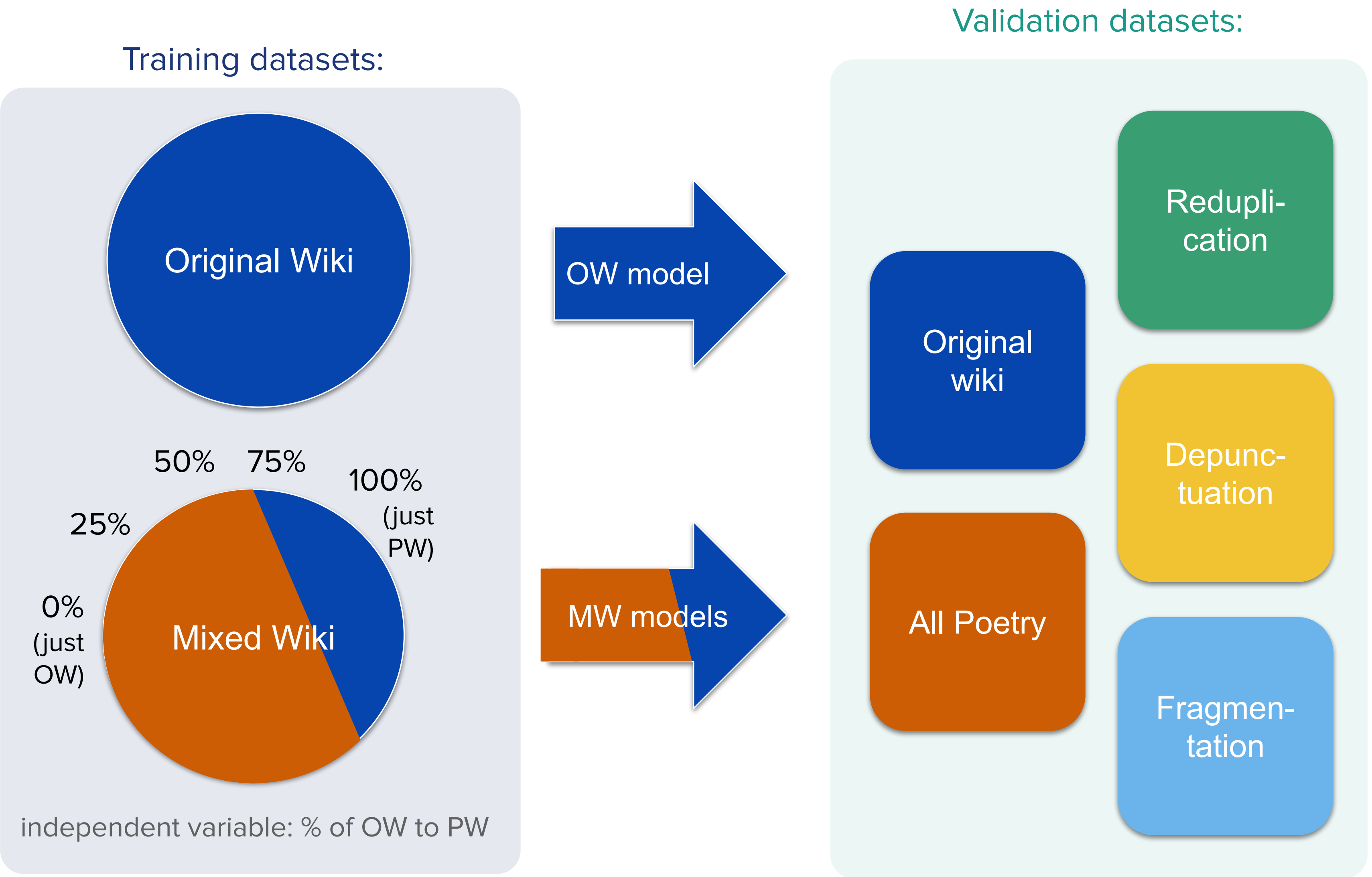
   c. De-punctionation

| Poetry Wiki | Vermont | is | a | state | in | the | New | England |
|---|---|---|---|---|---|---|---|---|
| Poetry tags | NNP | VBZ | DT | NN | IN | DT | NNP | NNP |

   d. All mutations combined (All Poetry)

| Poetry Wiki | Vermont | is | a | state | . | \n | in | the | New | New | England |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Poetry tags | NNP | VBZ | DT | NN | . | \n | IN | DT | NNP | NNP | NNP |

3. Train models using a various split of Original Wikipedia and All Poetry in the training data. Test each of these models on the Original Wikipedia & various poetry mutation datasets.


Validation datasets:

Training datasets:

Original Wiki

OW model

50% 75% 100% (just PW)

25%

0% (just OW)

Mixed Wiki

MW models

Original wiki

All Poetry

Redupli-cation

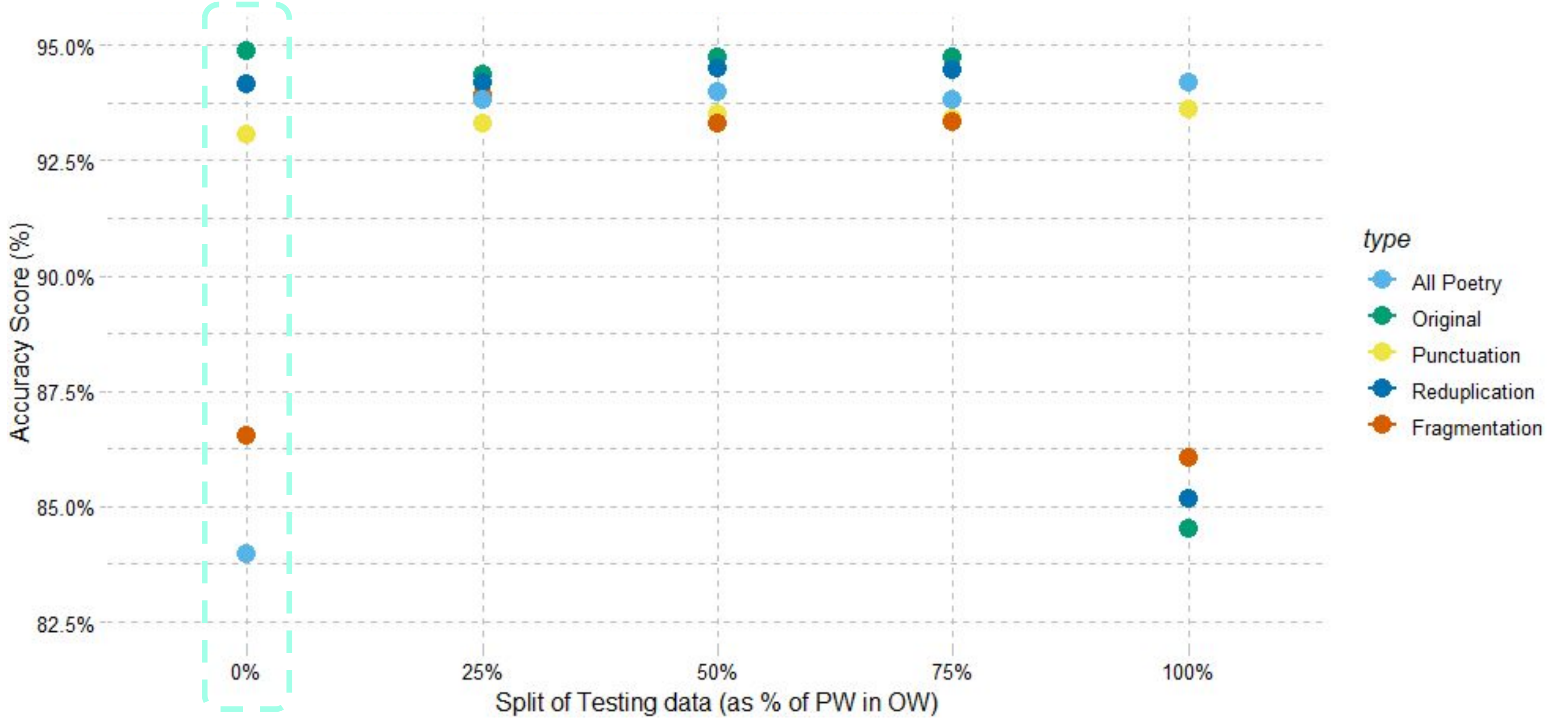
Depunc-tuation

Fragmen-tation

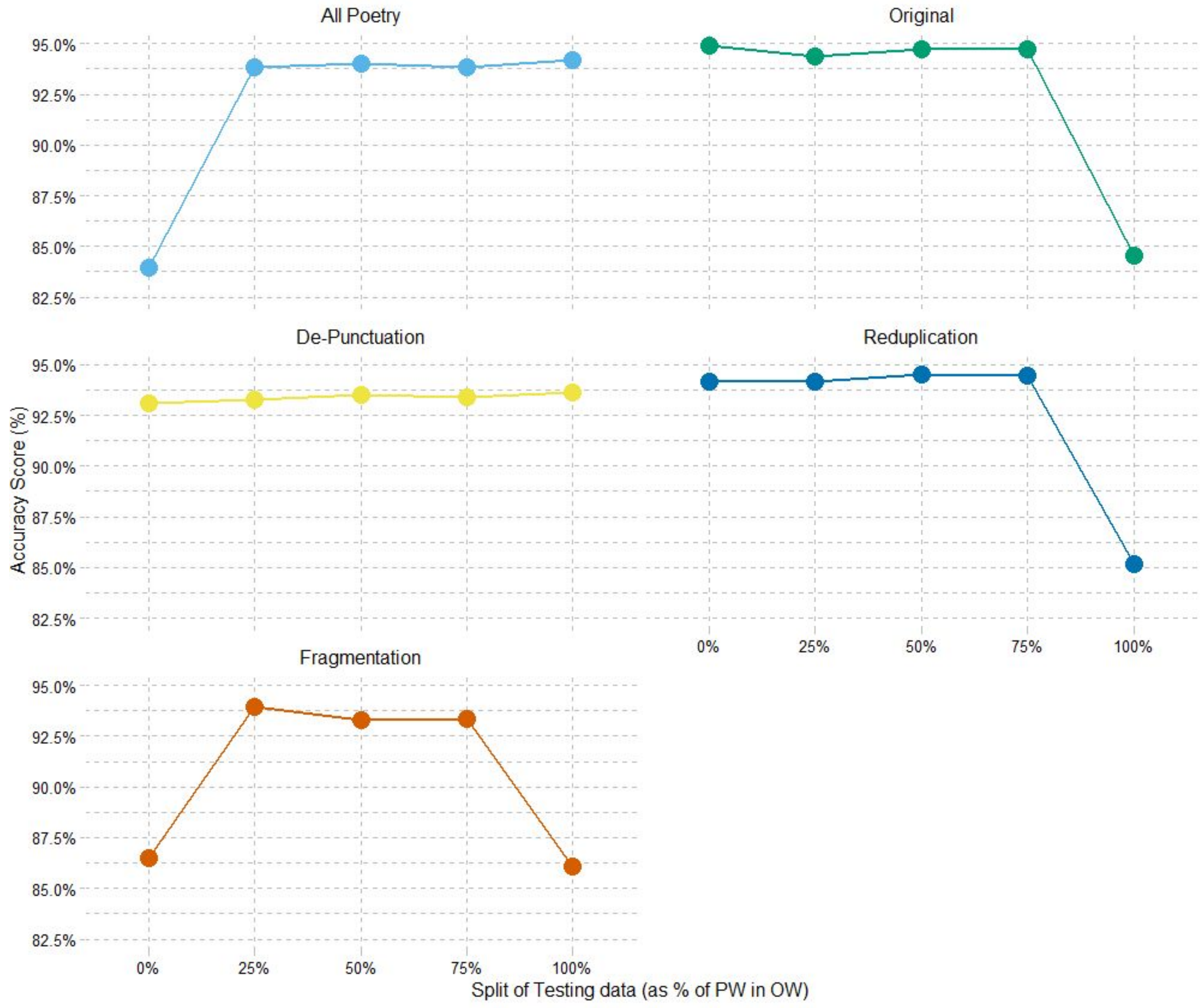
independent variable: % of OW to PW

We used a BiLSTM model architecture from the PyTorch and PyText library.[5] We used pretrained GloVe embeddings (400k vocab, 100 dimensions)[4] for our input dimensions, and the 36 fine-grained Penn Treebank tags[6] plus punctuation etc. (51 total) for the output dimensions. The 6 million English Wikipedia[2] articles were from the huggingface dataset collection, and the off-the-shelf POS tagger we used was SpaCy en_core_web_sm.

## Results



The accuracy scores of each model against each test data type. First column represents the baseline performance of the model trained only on Original Wikipedia (0% poetry).



For the Original Wikipedia test data, our results show that as long as some original text is in the model, the accuracy is comparable to the baseline 94.9%. Our results for each mutation are as follows:

a. **Fragmentation:** Fragmenting sentences causes confusion for both extremes of the training sets; our models score at most 93% only when both Original and Poetry Wikipedia is present in the model training set.
b. **Reduplication:** Our reduplication accuracy pattern follows that of Original Wikipedia.
c. **De-punctionation:** Getting rid of punctuation in the testing data is affected little by the proportion of original vs poetry data in the training data, which is intuitive if we consider that the lack of punctuation did not affect our sentence-break creation methods.
d. **All mutations combined (All Poetry):** Similar to the 0%, our models score around 95% as long as some Poetry Wikipedia is in the dataset.

Data augmentation is an effective and efficient method of mimicking syntactic conventions of poetry to create training data for machine learning models.

## Future Works

An extension of this project might address the limitations of applying data-augmentation methods to a text sequence. We are also interested in more tactical methods of poetry mutation, such as reduplicating or rearranging the order of text, being aware of clauses and other meaningful constituents of words. Accounting for conventions like rhythm and rhyme would improve applications to real-life poetry.

Extra special thanks to Prof. John Foley and Prof. Sayaka Abe for advising on my projects.
Thank you to Prof. Caplan and my fellow thesis class for their support over two great semesters.
I'd like to acknowledge Amy Yuen and Midd ITS faculty for their work on the Ada HPC.
Finally, to the rest of the CS department and my Midd friends, thank you for supporting my journey!

Check out my code at https://github.com/hjarai/pos and the full paper 📙 here →

## Works Cited

1. Xiang Dai and Heike Adel. An analysis of simple data augmentation for named entity recognition. arXiv preprint arXiv:2010.11683, 2020.
2. Wikimedia Foundation. Wikimedia downloads. URL: https://dumps.wikimedia.org
3. D. Jurafsky and J.H. Martin.Speech and Language Processing: *An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition Prentice Hall series in artificial intelligence.* Pearson Prentice Hall, 2009. URL: https://books.google.com/books?id=fZmj5UNK8AQC
4. Christopher D Manning. Part-of-speech tagging from 97% to 100%: is it time for some linguistics? *In International conference on intelligent text processing and computational linguistics*, pages 171–189. Springer, 2011
5. Ben. Trevett. *Pytorch pos tagging*. https://github.com/bentrevett/pytorch-pos-tagging, 2019.
6. Beatrice Santorini. Part-of-speech tagging guidelines for the penn treebank project. 1990.