**Abstract**

**POS tagging enhancement to NLP classification tasks**

**Omri Maoz, Tal Levi**

Tel-Aviv University

{omrimaoz, tallevi}@mail.tau.ac.il

In our research *we tested the hypothesis that a task of classifying contextualize phrases can be better execute with common NLP tools when feeding them with their corresponding word-to-word tagging phrases. The key observation when examine a contextual phrase is that there’s a common semantic between some words in phrases that are belong to the same class. For example, a positive review of a movie contains word that hold a positive semantic meaning and this semantic is common in words in other positive reviews. We believe that these semantic might be express with some POS tagging pattern in the corresponding POS tag phrases. Moreover, the POS tagging phrases significantly reduce the vocabulary size, which may allow to form patterns with fewer data (collect large, labeled data is often a problem itself). Therefore, we assume two assumptions to test in this project:*

1. *It possible to learn POS tagging pattern to achieve better result on NLP classification task.*
2. *An equivalate results can be achieve with fewer data*
3. **Introduction**

Classification of commercialize text phrases is a task with many applications that we use and affect our daily lives. An example is a technology that scan comments and reviews of users in social network, searching for violation of the terms and rules set by the network to remove and block these “bad” contents.

Table

Description automatically generatedA lot of work and study has been done in this subject and found that deep learning tools achieve good result. Models such as CNN-GRU and BiLSTM for examples achieve around 80% accuracy on Movie Reviews dataset with more than 10,000 reviews divided to 2 classes.

Table 1. Some common deep learning architectures accuracy results. (Source: [[1](#link1)])

Improving results for those NLP models is not an easy task and there’s many variables and parameters to consider alongside many approaches to test and experiment. One approach, which supported by recent research shows that preprocessing techniques of the data manage to improve NLP models results.

Part-of-Speech (POS) tagging is a popular NLP process which refers to categorizing words in a text in correspondence with a particular part of speech, depending on the definition of the word and its context.

Recent study on question classification in Thai language have shown that text classification can benefit from Part-of-Speech (POS) tagging as part of feature selection method to achieve better result [[2](#link2)]. In Thai language some words have different meaning when considered alone than when joining other words because of a meaning word based on ordering the sequence of words and context. Therefore, considering a syntactic feature for the obvious classification of Thai sentences is necessary. One feature the research editors manage to find by analyzing frequency of POS tag for each question type in the packages “PyThaiNLP” and “Stanford CoreNLP”, is that each question type was sensitive to some POS tagging ratios.

To test our research hypothesis, we focused on 3 NLP models – DAN, BiLSTM and Transformer.

We fed those models with 2 datasets:

1. “IMDB reviews” dataset, divided into 2 classes (positive, negative), with more than 15k reviews.
2. “News headline” dataset, divided to many topics as classes, but we focused on the 5 most common, with more than 3000 headlines per category.

One more motivation to use POS tagging as English words categories is that this preprocessing methodology is general and not depends on pre known knowledge so it can be applied to any given dataset.

1. **Related Work**

[Text classification](https://www.projectpro.io/project-use-case/nlp-text-processing-classification-python) is the process of classifying or categorizing the raw texts into predefined groups. There are many studies and works developing and testing numerous approaches and methods to tackle this problem and with great results. From fundamental machine learning algorithm such as linear regression and SVM to state-of-the-art deep learning models.

Today, commercial hi-tech companies integrate text classification technologies in their products. From filtering spam emails, to analyzing politician’s speeches, abuse content marking and removal and much more.

1. **Methodology**
   1. **Datasets**

For performance evaluation of the suggested hypothesis, we used two datasets to test binary classification and multi-class classification. The IMDB reviews dataset contain various length reviews dividing to positive and negative reviews. This dataset is interesting to extract feature from because positive and negative writing have many language forms (cynicism, exaggeration, straight-forward, etc.). The second dataset is the News headlines dataset, which contain 40 different subjects (as classes). In contrast to the first dataset, this dataset has approximately similar headlines length, but may contain some interesting tagging patterns. We chose 5 subjects to focus on (Politics, Entertainment, Wellness, Travel, Sport), each with more than 3k headlines. We used different sizes of subset from the datasets, in decrease order to test the 2-research assumption. Split validation factor set to be 0.2 for evaluation, and we preserved uniform distribution between classes.

* 1. **Data preprocessing**
     + - 1. **Tagging and Tokenization**

We used Stanza (Python NLP package) to create a corresponding POS tagging sentences for each sentence from the dataset by calculate each word it’s tag.

To tokenize words, we used spaCy (Python NLP Library) with ‘en\_core\_web\_sm’ trained pipeline.

* + - * 1. **Feature selection**

The goal is to improve text classification with the 3 tested NLP models. To do that we proposed and calculated 3 preprocess methods to produce features which in turn was fed to the models and tried to improve result accomplish by them.

The methods:

1. **Tag Filtering** (refer as filter) – when overviewing the text in the dataset we notice that some tags are more capable to form a combination with other tags to express a tag pattern and semantic relation. POS tags such as PUNCT (punctuation), PRON (pronoun), etc. are less likely to belong to a key subsentence which point to the semantic class of the sentence. On the other hand, NOUN, ADV (adverb), ADJ (adjective) and VERB are more likely to do so.

With “Tag Filtering” we remove all tags but NOUN, ADV, ADJ and VERB tags from the dataset.

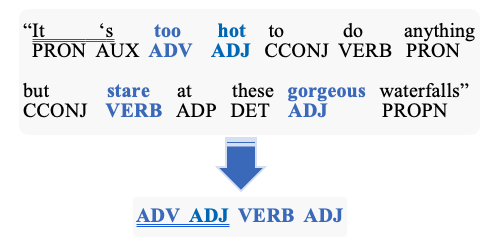


Figure 1. ‘Travel’ headline from the News headline dataset and it’s corresponding POS tag sentence.

1. Timeline

   Description automatically generated with medium confidence**Tag Extend** (refer as extend) – most of NLP deep-learning models work better with big vocabulary. By looking on POS tag corresponding phrases we greatly reduce the vocabulary size. To compensate this negative effect, we extended the 4 “interesting tags” (NOUN, ADV, ADJ and VERB) to subcategories.
   * NOUN and VERB – python NLP package call NLTK with WordNet lexical database by Princeton provide subcategories for nouns and verbs. We use Synset, a tool from NLTK to look for semantic words relation in WordNet, to distribute words to these subcategories.
   * ADJ and ADV – For ADJ and ADV we used the most common adjective and adverb subcategories in the English language. In addition, we added suffixes as subcategories and used Spacy similarity feature tool to distribute words with these subcategories.

Diagram

Description automatically generatedWe then concatenated the most 2 similar subcategories to the POS tag of any relevant word in the dataset.

Figure 2. Calculation pipeline to extend word tag according to the most similar subcategories. Example for VERB categorized word.

1. **Bigram Interpolation (bigram)** – In order to highlight tag patterns in text we used pairing 2 consecutive words in the sentence in a most common appearance in the dataset, with a decrease order. Doing that, we are emphasize common tag pairs over lesser common tag pairs.

Figure 3. Example of “Bigram Interpolation” on ‘Entertainment’ headline from the News dataset.

Lastly, we used different combination of the methods above as feature selection to test our models with.

* 1. **Classification (models)**

To test our hypothesis, a comparison needs to be done and for that we need relevant NLP models. DAN, BiLSTM and Transformer models are a good choice because while differ in their architecture, they are taking into calculation information between neighbor words in the text.

* + - * 1. **DAN – Deep Averaging Network** [[3](#link3)] **fill**
        2. **BiLSTM – Bidirectional Long Short-Term Memory** [[4](#link4)][[5](#link5)]

We constructed embedding layer of size 300 follow by 3 recurrent layers with 128 BiLSTM cells, a 50% dropout layer, a linear layer from 128 to 32 neurons with Relu activation function, and a Linear layer with Softmax activation function from 32 to the number of dataset classes neurons.

* + - * 1. **Transformer** [[6](#link6)][[7](#link7)]

Build with Encoder and Decoder with d\_model (expected feature) set to 32, heads to 4 and 2 hidden layers, followed by Linear layer with Softmax activation function from d\_model (32) to the number of dataset classes neurons.

Positional Encoding - contain embedding layer of vocabulary size follow by the traditional “Vaswani et al” function for constant position-specific values

1. ‘bigram’ (Tag preprocessing) use UPOS tags. The different between “(2) upos and filter” and “(5) bigram and filter” for example is only with the use of “Bigram Interpolation” or not.

where refer to the position along the embedding vector and refer to the order of the word in the sentence.

Encoder – Contain 2 consecutive Norm layers, followed by a Multi-Head-Attention layer and a Feedforward layer.

Decoder – Contain 3 consecutive Norm layers, followed by 2 consecutive Multi-Head-Attention layers and a Feedforward layer.

1. **Evaluation – Metrics**

To evaluate results, we chose Accuracy and F1-score metrics. Accuracy is the measure of all the

correctly identified cases. At the end, the task is classification, so we interest with classify correctly. F1-score is the harmonic mean of Precision and Recall, therefore gives a better measure of the incorrectly classified cases than the Accuracy metric. We used F1-score to determine how balance our models preform.

1. **Experiments**

In the proposed approach, we studied the effect of using POS tags on datasets for comparing the various data preprocessing tasks described in the previous part including (1) upos, (2) upos and filter, (3) upos, filter and extend, (4) bigram, (5) bigram[1](#comment1) and filter, (6) bigram[1](#comment1) and extend, (7) bigram[1](#comment1), filter and extend.

We tested our experiments with the 3 models. As evident from Table 1, the comparison results on the Accuracy score considering some feature selection with 2 out of 3 models could increase performance.

1. **Conclusions**

 Looking on POS tags and extract POS tags patterns from dataset phrases can improve classification results for some models. From the experiments results, we see a big improvement with the binary-class dataset over the multi-class dataset. One possible explanation for that might be that in the News headline dataset, in addition for it being a multi-class (more difficult task), each phrase in the data was considerably shorter that the ones in the IMDB reviews dataset. This may have affected the overall performance of extracting POS tags patterns.

By examining the F1-score (relative to the Accuracy) from the results, we can tell that the model’s performance was balanced which is the desirable behavior.

Regard the second assumption in the beginning of this paper, we see that with the BiLSTM model, with bigram-filter preprocessing we achieve better performance than the original data even with significantly small dataset.

Lastly, at least with 2 out of the 3 models, we see that the bigram and bigram-filter preprocessing methods achieved better results than other approaches. It makes sense these methods worked. The bigram highlights common word pairs over less common pairs, and the filter may reduce noise over the dataset (also explain why it wasn’t effective with the News dataset – It didn’t have much noise to reduce.

**References**

1. [D. Raihan. Deep Learning Techniques for Text Classification. Towards Data Science. April 2021.](https://towardsdatascience.com/deep-learning-techniques-for-text-classification-78d9dc40bf7c)

1. [S. Chotirat and P. Meesad. Part-of-Speech tagging enhancement to natural language processing for Thai wh-question classification with deep learning. Heliyon, volume.7, issue 10. October 2021.](https://reader.elsevier.com/reader/sd/pii/S2405844021023197?token=3F74C80B869405C09F48CEEAB845BF5CAD8662BDC4DEB284BD887D1CE7DE091AE92E4919FB0564D2C0F63444953AC1C2&originRegion=eu-west-1&originCreation=20220422224829)

1. [M. Iyyer. V. Manjunatha. J. Boyd-Graber. H. Daume ́ III. Deep Unordered Composition Rivals Syntactic Methods for Text Classification. Association for Computational Linguistics. July 2015.](https://people.cs.umass.edu/~miyyer/pubs/2015_acl_dan.pdf)

1. [R. Cheng. LSTM Text Classification Using Pytorch. Towards Data Science. July 2020.](https://towardsdatascience.com/deep-learning-techniques-for-text-classification-78d9dc40bf7c)

1. [A. NS. Multiclass Text Classification using LSTM in Pytorch. Toward Data Science. April 2020.](https://towardsdatascience.com/multiclass-text-classification-using-lstm-in-pytorch-eac56baed8df)

1. [S. Lynn-Evans. How to code The Transformer in Pytorch. Toward Data Science. September 2018.](https://towardsdatascience.com/how-to-code-the-transformer-in-pytorch-24db27c8f9ec)

1. [A. NS. Multiclass Text Classification using LSTM in Pytorch. Toward Data Science. April 2020.](https://towardsdatascience.com/multiclass-text-classification-using-lstm-in-pytorch-eac56baed8df)