Title

Tal Levi, Omri Maoz

Tel Aviv University

Abstract

We were to test the hypothesis that a task of classifying contextualize phrases can be better execute with common NLP tools when feeding them with their word to word corresponding 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 phrase. Moreover, the POS tagging sentences significantly reduce the vocabulary size, therefore we assume two assumption 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 application that we use and affect our daily lives. For 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 In order to remove and block those “bad” contents.

A 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 achive around 80% accuracy on Movie Reviews dataset with more then 10,000 reviews divided to 2 classes.

Recent work have shown that a task of Question classification in Thai language

Abstract

Abstract

We were to test the hypothesis that a task of classifying contextualize phrases can be better execute with common NLP tools when feeding them with their word-to-word corresponding 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 phrase. Moreover, the POS tagging sentences significantly reduce the vocabulary size, 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 in order to remove and block those “bad” contents.

A 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.

Add link or image from the site (a work not a study)- <https://towardsdatascience.com/deep-learning-techniques-for-text-classification-78d9dc40bf7c>

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. 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.

Add link to the study.

In order to test our experiment hypothesis, we are focusing on 3 NLP models:

* DAN – Deep Averaging Network.
* BiLSTM – Bidirectional Long-Short Term Memory
* Transformer.

We will feed those models with 2 datasets:

* “IMDB reviews” dataset, divided into 2 classes (positive, negative), with more then 15k reviews.
* “News headline” dataset, divided to many topics as classes, but we will focus on the 6 most common, with more than 3000 headlines per category.

One more motivation to use POS tagging as general 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.

We organize the rest of the paper as follow. פסקה על המשך המאמר – ךהשלים בסוף

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 hypothesis suggested 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 News headlines dataset, which contain 40 different subjects (As classes). In contrast to the first dataset, this dataset has approximately close headlines length, but may contain some interesting tagging patterns. We chose 7 subjects to focus on (Politics, Entertainment, Wellness, Travel, Sport, Busyness and Crime), each with more than 3k headlines. We will use different sizes of subset from the datasets, with decrease order in order to test the 2-research assumption, split validation factor will be 0.2 for evaluation, and we will preserve uniform distribution between classes.

* 1. Data preprocessing
     1. Tagging and Tokenization

We used Stanza (Python NLP package) to create for each sentence from the dataset a corresponding POS tagging sentences 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. In order to do that we calculate 3 preprocess methods to produce features which in turn will be fed to the models and will try to improve result accomplish by them.

The methods:

## Tag Filtering (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 PUNCT (punctuation), PRON (pronoun), etc. are less likely to belong to a key sub-sentence which point to the class semantic of the sentence. On the other hand, NOUN, ADV (adverb), ADJ (adjective) and VERB are more likely to do so. For example, a ‘Crime’ headline from the News headline dataset: Add figure from below

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

1. Tag Extend (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. In order to compensate this effect, we will extend the 4 “interesting tags” (NOUN, ADV, ADJ and VERB) to subcategories.
   1. NOUN and VERB – python NLP package call NLTK with WordNet lexical database by Princeton are providing with subcategories for nouns and verbs. We use Synset, a tool from NLTK to look for semantic relation words in WordNet, to distribute words to the subcategories.
   2. ADJ and ADV – We took ADJ and ADV subcategories from (add-links), add suffix as subcategories and used Spacy similarity feature tool to distribute words with these subcategories.
2. Bigram Interpolation (bigram)– In order to highlight tag patterns in text we used pairing 2 close words in the text 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.

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

* 1. Classification (models) – In order to test our hypothesis a comparison need 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: *fill*
     2. BiLSTM – Bidirectional Long Short-Term Memory - 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 dataset classes number neurons.
     3. Transformer – 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 dataset classes number neurons.
     4. Positional Encoding - contain embedding layer of vocabulary size follow by the traditional Vaswani et al function for constant position-specific values -, where refer ti 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

For 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 classified 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 will use 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 and filter, (6) bigram and extend, (7) bigram, filter and extend.

We tested our experiments with the 3 models describe on part 3.

As evident from [Table 1](https://www.sciencedirect.com/science/article/pii/S2405844021023197" \l "tbl0040), the comparison results on the Accuracy considering some feature selection with 2/3 models could increase performance. Add note that bigram is upos based.

1. Conclusions:
   1. As for the first research hypothesis, looking on POS tags and extract POS tags patterns from dataset phrases can improve classification results for some models. From the experiment 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 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 dataset. This may have affected the overall performance of extracting POS tags patterns.
   2. Regard the second research hypothesis, we see that in the BiLSTM model, with bigram-filter preprocessing we achieve better permormance than the original data, and even with significantly small dataset. Note: add why we have chosen this question, and why the answer is important
   3. 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 noise to reduce.

“It ‘s **too** **hot** to do anything

PRON AUX **ADV** **ADJ** CCONJ VERB PRON

but **stare** at these **gorgeous** waterfalls”

CCONJ **VERB** ADP DET **ADJ** PROPN

**ADV** **ADJ**  **VERB** **ADJ**

1. Introduction – general talk, dataset, modeling, what we attend to do
2. Related work - <https://www.sciencedirect.com/science/article/pii/S2405844021023197>

<https://towardsdatascience.com/deep-learning-techniques-for-text-classification-78d9dc40bf7c>

https://medium.com/analytics-vidhya/accuracy-vs-f1-score-6258237beca2

גם לקחת השראה מכאן

1. Methodology – models, data preprocessing, tokenization, embedding, feature selection
2. Evaluation – metrics
3. Experiment result
4. Conclusions

**ADV** **ADJ**  **VERB** **ADJ**

“It ‘s **too** **hot** to do anything

PRON AUX **ADV** **ADJ** CCONJ VERB PRON

but **stare** at these **gorgeous** waterfalls”

CCONJ **VERB** ADP DET **ADJ** PROPN

“The 5 best new Netflix

DET NUM ADJ ADJ PROPN

shows of April 2018”

NOUN ADP PROPN NUM

3.24%

0.11%

“The 5 best new Netflix shows of April 2018”

DET NUM ADJ ADJ PROPN NOUN ADP PROPN NUM

A picture containing black, domestic cat

Description automatically generatedA picture containing black, domestic cat

Description automatically generated

A picture containing black, domestic cat

Description automatically generatedA picture containing black, domestic cat

Description automatically generatedA picture containing black, domestic cat

Description automatically generatedA picture containing black, domestic cat

Description automatically generatedA picture containing black, domestic cat

Description automatically generatedA picture containing black, domestic cat

Description automatically generatedA picture containing black, domestic cat

Description automatically generatedA picture containing black, domestic cat

Description automatically generated

0.42%

1.69%

3.83%

0.11%

3.24%

0.85%

0.38%

0.12%

A picture containing black, domestic cat

Description automatically generatedA picture containing black, domestic cat

Description automatically generatedA picture containing black, domestic cat

Description automatically generated

0.85%

0.38%

0.12%

1.69%

0.42%

A picture containing black, domestic cat

Description automatically generatedA picture containing black, domestic cat

Description automatically generated

A picture containing black, domestic cat

Description automatically generatedA picture containing black, domestic cat

Description automatically generated

3.83%

Word: “**Shows**”

UPOS: **VERB**

UPOS: **VERB\_communication\_perception**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | | | DAN | | | BiLSTM | | | Transformer | | | |
| Prepro-cessing method | Dataset | Metrics | 15000 | 5000 | 1000 | 15000 | 5000 | 1000 | | 15000 | 5000 | 1000 |
| original | IMDB | Acc  F1 | 0.839  0.839 | 0.83  0.835 | 0.855  0.853 | 0.751  0.749 | 0.65  0.645 | 0.6  0.65 | | 0.756  0.861 | 0.672  0.865 | 0.75  0.857 |
| News | 0.858  0.64 | 0.782  0.78 | 0.805  0.812 | 0.6  0.76 | 0.51  0.66 | 0.45  0.55 | | 0.841  0.643 | 0.792  0.746 | 0.688  0.56 |
| upos | IMDB | 0.628  0.626 | 0.612  0.608 | 0.655  0.655 | 0.584  0.605 | 0.59  0.63 | 0.58  0.61 | | 0.752  0.859 | 0.5  0.5 | 0.5  0.5 |
| News | 0.493  0.349 | 0.418  0.407 | 0.47  0.462 | 0.363  0.389 | 0.45  0.505 | 0.41  0.51 | | 0.714  0.48 | 0.65  0.45 | 0.643  0.4 |
| upos-filter | IMDB | 0.554  0.554 | 0.549  0.533 | 0.595  0.566 | 0.54  0.55 | 0.56  0.56 | 0.56  0.57 | | 0.751  0.858 | **0.753**  0.859 | 0.764  **0.866** |
| News | 0/425  0.323 | 0.315  0.309 | 0.32  0.32 | 0.361  0.23 | 0.31  0.35 | 0.31  0.41 | | 0.679  0.43 | 0.65  0.44 | 0.63  0.36 |
| upos-filter-extend | IMDB | 0.555  0.555 | 0.566  0.566 | 0.52  0.516 | 0.73  0.74 | **0.69**  **0.705** | 0.59  0.61 | | 0.751  0.858 | **0.758**  0.863 | 0.76  **0.863** |
| News | 0.377  0.278 | 0.268  0.216 | 0.215  0.176 | 0.455  0.5 | 0.38  0.45 | 0.32  0.39 | | 0.772  0.555 | 0.66  0.51 | 0.64  0.41 |
| bigram | IMDB | 0.622  0.621 | 0.607  0.602 | 0.64  0.645 | **0.857**  **0.852** | **0.75**  **0.745** | 0.635  0.645 | | 0.756  0.861 | **0.754**  0.86 | 0.75  0.857 |
| News | 0.492  0.342 | 0.417  0.412 | 0.48  0.478 | 0.493  0.6 | 0.51  0.61 | **0.49**  **0.6** | | 0.71  0.48 | 0.68  0.55 | 0.65  0.45 |
| bigram-filter | IMDB | 0.555  0.552 | 0.552  0.534 | 0.595  0.579 | **0.96**  **0.96** | **0.96**  **0.96** | **0.925**  **0.925** | | 0.753  0.858 | **0.752**  0.862 | 0.756  **0.861** |
| News | 0.417  0.323 | 0.323  0.316 | 0.3  0.297 | 0.373  0.245 | 0.39  0.44 | 0.39  0.45 | | 0.683  0.44 | 0.637  0.467 | 0.61  0.4 |
| bigram-extend | IMDB | 0.621  0.621 | 0.62  0.621 | 0.605  0.602 | 0.7  0.72 | **0.67**  **0.66** | 0.55  0.6 | | 0.751  0.858 | **0.751**  0.858 | 0.5  0.5 |
| News | 0.442  0.442 | 0.389  0.384 | 0.44  0.43 | 0.43  0.44 | 0.45  0.56 | 0.41  0.49 | | 0.79  0.58 | 0.688  0.55 | 0.61  0.638 |
| bigram-filter-extend | IMDB | 0.553  0.552 | 0.561  0.56 | 0.56  0.552 | 0.705  0.703 | 0.62  0.62 | 0.55  0.54 | | 0.751  0.858 | **0.755**  0.86 | 0.756  **0.861** |
| News | 0.378  0.378 | 0.271  0.215 | 0.25  0.222 | 0.41  0.41 | 0.29  0.34 | 0.22  0.29 | | 0.77  0.568 | 0.645  0.515 | 0.63  0.41 |

|  |  |  |
| --- | --- | --- |
|  | Model | Movie Reviews |
| 0 | BiLSTM-rand | 77.6 |
| 1 | BiLSTM-static | 79.5 |
| 2 | BiLSTM-dynamic | 79.8 |
| 3 | 1D CNN-static | 79.0 |
| 4 | 1D CNN-dynamic | 79.4 |
| 5 | Ensemble CNN-GRU-rand | 77.0 |
| 6 | Ensemble CNN-GRU-static | 79.8 |
| 7 | Ensemble CNN-GRU-dynamic | 79.4 |