CENG 783 – Deep Learning
Progress Report

DEEP LEARNING VIEW TO NATURAL LANGUAGE PROCESSING

Aybars Kerem TASKAN, M.S.
Department Of Computer Engineering
Middle East Technical University
Ankara, Turkey

***Abstract*—Before deep learning makes very important steps in the areas of speech recognition and language translations, natural language processing were being used to achieve success in these areas. However, deep learning(DL) has become superior to natural language processing(NLP) for some part of these areas where dataset is large. This paper describes the transition procedure from NLP to NLP with DL and makes a systematic comparison between these two methods.**

***Keywords*—Deep Learning, Natural Language Processing**

1 Introduction

In this paper, I am going to explain a few intuitive methods to show how some of the current natural language processing methods would be implemented in deep learning(DL) with a higher level of success for some problems and the reasons why deep learning with Natural Language Processing(NLP) is able to solve these problems better than NLP alone.

I will firstly use only NLP and then NLP with Deep Learning to get my results. Then, I will make comparisons between

2 Discussion of the Literature:

2.1 Literature Survey:

There are papers published on mentioning trends on NLP with Deep Learning [1], applying Deep Learning on NLP [2] using CNNs. There are even papers applying very deep learning for NLP tasks[3].

2.2 Problem definition:

To the best of my knowledge**,** applications proposed are generally apply DL methods with NLP methods hand to hand while training and testing the inputs(datasets). However, to apply deep learning on NLP, we generally have to change our structure from scratch or carefully design the overall architecture to apply NLP and DL together.

2.3 Novelty:

If there were a way to apply DL approach for NLP to the results we obtained from NLP directly, a DL expert without a deep knowledge about NLP, would be able to conduct a NLP with DL experiment given an NLP output.

In this paper, I will explain how to apply Deep Learning methods to "term frequency vector" for text classification/analysis and other corresponding vector representations for applications other than text classification.

3 Explanation of the Dataset

I used a deep-nlp dataset from kaggle: https://www.kaggle.com/samdeeplearning /deepnlp/downloads/deepnlp.zip/1.

My dataset includes 80 responses from different people to the question "**Describe a time when you have acted as a resource for someone else**".

There are two resulting classes for the responses **flagged** and **not_flagged**

If the explanation of the user seems enough to cover the scope of the question, it is marked as **flagged**. However, if it seems that the answer should be expanded to, really, answer the question we mark/classify it as **not_flagged**.

The classification as flagged and not_flagged has been done by a human and therefore we take this classification as the ground truth for our tests.

4 Methodology

My task composes of two main parts: Natural Language Processing(NLP) and Deep Learning(DL).

The NLP part of my task with the methodologies and tools I used for NLP can be found in section 5.

For the DL part of my task,

I firstly plan to use a basic neural network for sentence modeling by extracting high level features from unigrams resulted from my NLP task.

I secondly plan to use a convolutional neural network since CNNs are good at extracting position invariant features from a given input[4]. The length of the text is not important for me since I will focus on text classification instead of question answering and translation where long sentence semantic and position of the words are important.

Later, I will compare the position invariant features I extracted for my train cases with my test cases. According to whichever class(flagged,not_flagged) the test text features resemble most, I will find out the classification result.

After adding DL methods to my NLP, I will compare the accuracies of my NLP classifier and NLP with DL classfier for challenging cases.

Lastly, I will define the transition prodecure from NLP to NLP with DL as a function of the NLP output.

5 INITIAL RESULTS:

My initial results belong to NLP part of my project. For the DL part, I extracted the n-grams as unigrams, resulting vector structure of my NLP task; but I plan to use digrams for better representation of my inputs as well. Moreover, my studies for the DL with NLP is finished and implementation of the DL part will be done using the output vectors of NLP.

For the implementation of DL part, I plan to extract high level features of the unigrams I found and feed them into my neural network. Then I will choose appropriate filters for my convolutional networks.

5.1 Tools and Studies

Until now, I studied the NLP methods and how to apply NLP on inputs given a dataset. I studied the applications of DL on NLP from various sources. I searched for libraries used in the field and necessary library functions to use with their whys and whens. So far, I have finished applying NLP on my dataset and therefore get the initial inputs for the deep learning task since output vector structure of my NLP also means input for my deep learning task.

The task-related tools I used are:

numpy, panda, workcloud, nltk, matplotlib,

I used 4 types of modules and subtools which are "stemmers", "lemmatizers", "visualizer for words" and "word type distinguisher"

Stemmers: nltk.stem.snowball.SnowballStemmer

I prefer 'english' over 'porter' as type of my stemmer since english version stems words, less keeping its meaning better.

Lemmatizers: nltk.wordnet**.**WordNetLemmatizer.

After both trying lemmatizer and stemmer, lemmatizer seems to fit for my purpose better since the words do not lose its meaning and structure after lemmatizing unlike stemming.

Visualizers for Words: wordcloud.WordCloud

I used it for visualization of words according to their frequency. It helps understanding the data better.

Word Type Distinguisher: nltk.pos_tag

I used to distinguish verbs, adjectives, nouns etc. from each other. I used this module to extract only the nouns from my text since I consider the other types as unrelated for my purpose.

5.2 Methodology and the Conduction of the experiment:

Test set is obtained by collecting answers other than the initial 80, to the question mentioned in section 3; and using the first 80 responses as the training set, we should guess the classification of the new answers as "flagged" or "not_flagged".

I used top 15 frequent nouns as the classification factor for each text and classified the new response according to the 80 initial responses.

I used the **sum of the multiplications** technique for the same nouns.(I could have used sum of squares to make the difference larger, but this seemed enough for my experiment.)

For example, for the example sentence,

"Once I helped my roommate to catch the school bus by stopping it", the nouns are:

-> {'help': 1, 'roommate': 1, 'school': 1, 'bus': 1, 'stop':1 }

For the train text of flagged responses(25 flagged classes out of the first 80 initial responses), the top 15 frequent words that we use to compare are:

-> {'FRIEND': 25, 'HELP': 15, 'TALK': 11, 'PEOPLE': 11, 'TRY': 8, 'DEPRESSION': 7, 'SCHOOL': 7, 'YEAR': 6, 'ADDICTION': 5, 'LIFE': 5, 'ANXIETY': 5, 'ISSUE': 4, 'KILL': 4, 'GIRL': 3, 'SUICIDE': 3}

Computation is done like this: take the same words from both frequency vectors and multiply them. Then sum all them up. The same words that they both have are "help" and "school". Then,

For noun help: 1*15

For noun school: 1*7

Sum them up = 15 + 7 = 22 is our score for the flagged case

5.3 Intermediate NLP Text Classification Results:

For example out of the 80 initial responses,

The top 15 words for flagged texts are:

{'FRIEND': 25, 'HELP': 15, 'TALK': 11, 'PEOPLE': 11, 'TRY': 8, 'DEPRESSION': 7, 'SCHOOL': 7, 'YEAR': 6, 'ADDICTION': 5, 'LIFE': 5, 'ANXIETY': 5, 'ISSUE': 4, 'KILL': 4, 'GIRL': 3, 'SUICIDE': 3}

To visualize our frequent words, wordcloud of all flagged nouns are:

Figure 1: Flagged Frequent Nouns

The top 15 words for not flagged texts are:

{'FRIEND': 25, 'HELP': 19, 'TALK': 13, 'PEOPLE': 11, 'FEEL': 8, 'LISTEN': 8, 'SOMEONE': 7, 'PROBLEM': 7, 'TRY': 6, 'THINK': 6, 'NEED': 6, 'LIFE': 6, 'LOT': 6, 'GIRL': 5, 'CALL': 4}

To visualize our frequent words, wordcloud of all not flagged nouns are:

Figure 2: Not Flagged Frequent Nouns

Note that all letters are in upper case in the frequent words above and there are no words that have the same lemma due to filtering we made. Also all of them are nouns. If I chose lower case, then the words like "I" would be read as "i" and we would have included them as nouns since nltk.pos_tag accepts unknown words as nouns .

Default lemmatization works with nouns, therefore extracting only the nouns from my sentences is done before lemmatization. Also, I looked at the lemma of the words, if they are nouns I accepted them as nouns as well. Therefore, the lemma of word "helped" which is "help" is counted as a noun. It was necessary to keep the meaning.

When the new sentence added, we check whether the new sentence is similar to flagged sentences or not flagged sentences by comparing frequencies of their similar words. If the new sentence is more similar to flagged sentences we classify it as flagged; otherwise we classify it as not flagged.

5.4 Final NLP Text Classification Results:

For example in my experiment, two of the example sentences I added are:

- "Once I helped my roommate to catch the school bus by stopping it."

- "Once I helped friend to catch the school bus by by stopping it. Otherwise, he would have missed the class."

Since these sentences covers the answer well explaining how, I determined the ground truth as "flagged".

When I test 1st example, my NLP classifier give 22 and 19 for the scores of flagged and not flagged respectively.

When I test 2nd example, my NLP classifier give 47 and 44 for the scores of flagged and not flagged respectively.

Therefore the two sentences above are classified as **flagged** correctly.

The other example I added is:

"Once I helped someone. I remember it as if it happened today."

Since it does not explain how helping occurred well, I determined the ground truth as not flagged. The scores are 15 and 26 for the flagged and not-flagged tasks respectively. Therefore my NLP classifier classified this sentence as **not flagged** correctly.

References

[1] T. Young, D. Hazarika, S. Poria, E. Cambria 'Recent Trends in Deep Learning Based Natural Language Processing', 2018 [Online] Available:https://arxiv.org/pdf/1708.02709.pdf [Accessed April 7th 2018]

[2] Y. Kim 'Convolutional Neural Networks for Sentence Classification', 2014 [Online] Available:https://arxiv.org/pdf/1408.5882.pdf [Accessed April 7th 2018]

[3] A. Conneau, H. Schwenk, Y.L. Cun, L. Barrault 'Very Deep Convolutional Networks for Text Classification', 2017 [Online] Available: https://arxiv.org/pdf/1606.01781.pdf [Accessed April 8th 2018]

[4] W. Yin, K. Kann, M. Yu, H. Schutze 'Comparative Study of CNN and RNN for Natural Language Processing', 2017 [Online] Available: https://arxiv.org/pdf/1702.01923.pdf [Accessed April 9th 2018]