Reproducibility Project

CS598 Deep Learning for Healthcare
May 5, 2023

Contributors:
Michael Pettenato - mp34@illinois.edu
Adam Michalsky - adamwm3@illinois.edu

Scope of Reproducibility

We have implemented a reproduction of the experiment discussed in the original paper to validate the following claims:

1. A Stanford Parser and Word2Vec embedding approach can reduce model training time.
2. Convolution layers can perform sentence feature extraction reliably.
3. A CNN-based similarity analysis is competitive compared to similar research.

Note: All code was implemented from scratch based on the details shared in the original paper.

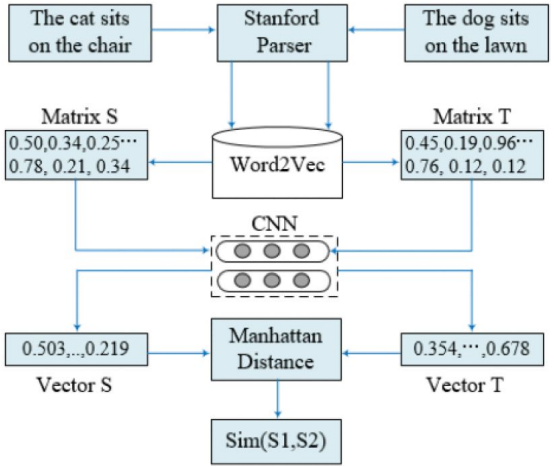
Overview

Approach consists of three stages:

- Data Pre-processing
- Neural Network Architecture
- Training and Results

About the data…

- Microsoft Research Paraphrase Corpus (link)
  - 5800 sentence pairs from new sources
  - Contains human annotations on similarity judgements

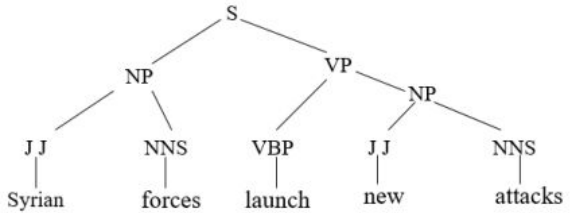


Data Preprocessing - Parsing

- Zhang et. al. 2019 leveraged a Stanford Parser (stanza) with SPO kernel extraction.
- SPO Kernel Functions - Functions that extracted primary features of a sentence (*subject*, *predicate*, & *object*)

Ablations:

1. New parsing methods:
    a. SpaCy - Alternative NLP package available (link).
    b. Raw - No SPO kernel function is used.
2. Concurrency & GPU support



Data Preprocessing - Embedding

- Embedding was done using Word2Vec
- Custom PyTorch Dataset implemented: *MSPCDataset*

Ablations:

1. Word2Vec modeling training
    a. Pre-trained model provided by gensim (trained on Google news vectors)
    b. Model trained on MSRP corpus dataset
    c. Fake embedding was introduced to account for vocabulary that is not found in the pre-trained model.

```
import gensim.downloader as api
pretrained_word2vec_path = api.load("word2vec-google-news-300", return_path=True)
print(pretrained_word2vec_path)
```

Neural Network Architecture

Dynamic K-Max Pooling

- Original paper used Dynamic K-Max Pooling as defined by [Kalchbrenner et. al 2014]

$$k = \max\left(k_{top}, \left\lceil \frac{L-l}{L}|s| \right\rceil\right)$$

Similarity Calculation

$$Man(\vec{V}_x, \vec{V}_y) = |x_1 - y_1| + |x_2 - y_2| + \ldots + |x_n - y_n|$$

$$score = e^{-Man(\vec{V}_x, \vec{V}_y)}, \quad score \in [0, 1]$$

Network Architecture

Layers	Configuration	Activation Function
Conv1d	input channel `embedding_dim` , output channel `num_filters`	ReLU*
DynamicKMaxPool	(k=3, l=1, L=3)	-
Conv1d	input channel `num_filters` , output channel `num_filters * 2`	ReLU*
KMaxPool1d	(k=3)	-
Linear	input = `k * num_filters *2` output = `hidden_dim`	Sentence Similarity

Training

Hyperparameters

- Batch Size: 64
- Number of Epochs: 80(Stanza-Pretrained used 20 epochs)
- Learning Rate: 1e-2
- Padding: 1
- Kernel: 3
- k_{top}: 3

Loss: Mean Square Error

Optimizer User: Stochastic Gradient Descent

Finding the Optimal Model Hyperparameters

A process of permuting hyperparameters with iterative training and testing runs was used to find optimal settings for the model

Model Type	# Epochs	Embedding_dim	Num_filters	Sent Vector size	lr	Training time	Accuracy	F1 score
Stanza-MSRP	80	50	150	300	0.01	1 min 55 s	0.6945	0.809
Stanza-Google	20	300	500	1000	0.01	1 min 58 s	0.708	0.816

Finding

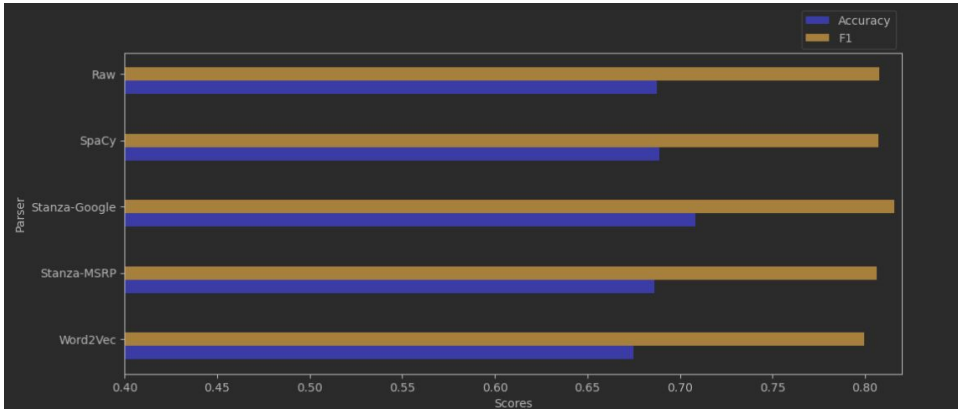
Sentence vectors produced by the CNN yielded better accuracy scores when they had more dimensions. We believe this is due to the sentence vectors being more descriptive in higher dimensions.

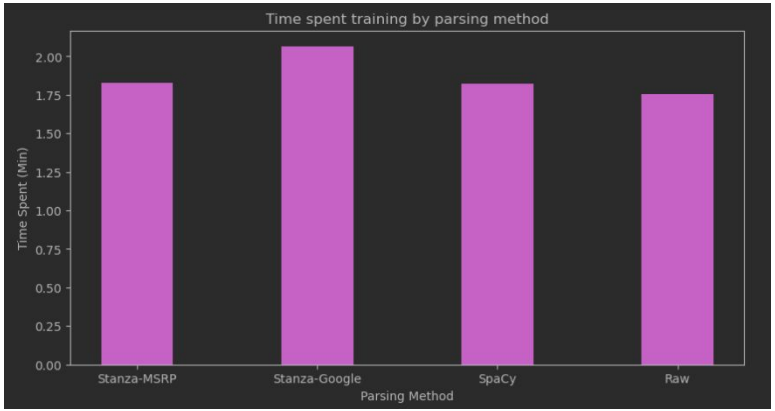
Note

The result of all of the permutations can be found in a file called **results/config-stanza-msrp.csv** in the github project repository and the code used generate the output of this file can be found in the **main.ipynb** jupyter notebook in a function called **find_training_parameters**

Results

Stanza-Google slightly outperformed the rest with an accuracy of *0.708* and a F1 score of *0.816*, but also had the longest training time.





Final Remarks

Claims

1. A Stanford Parser and Word2Vec embedding approach can reduce model training time.
    a. This approach out performed the alternative parsing methods. A pre-trained Word2Vec model added accuracy at the cost of additional training time.
2. Convolution layers can perform sentence feature extraction reliably.
    a. We achieved high accuracy scores (~ 0.68 +) using a convolutional neural network.
3. A CNN-based similarity analysis is competitive compared to similar research.
    a. The accuracy we achieved was similar to the accuracies of similar research documented in the original paper.

Links & References

Jupyter Notebook & Documentation: https://github.com/mikepettenato/cs-598-dl4health-final-project

References:

1. P. Zhang, X. Huang and M. Li, "Disease Prediction and Early Intervention System Based on Symptom Similarity Analysis," in IEEE Access, vol. 7, pp. 176484-176494, 2019, doi: 10.1109/ACCESS.2019.2957816.
2. Kalchbrenner, Nal, Edward Grefenstette, and Phil Blunsom, "A convolutional neural network for modelling sentences." *arXiv preprint arXiv:1404.2188* (2014).
3. Hua He, Kevin Gimpel, and Jimmy Lin “Multi-Perspective Sentence Similarity Modeling with Convolutional Neural Networks” Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1576–1586, Lisbon, Portugal, 17-21 September 2015. c 2015 Association for Computational Linguistics.

Future Ablations

A future ablations that may be worth considering is to compare the CNN model for sentence accuracy described in this paper with the CNN model described in “Multi-Perspective Sentence Similarity Modeling with Convolutional Neural Network” paper.

This paper describes a process whereby sentences are passed through sentence convolutions with different sized kernels , think of ngrams, and dimensional convolutions to try and extract interesting features from the embedding dimensions themselves.

The model leverages a combination of max, mean, and min polling layers.

It then combines these findings into a sentence vector representation and compares sentence-1 with sentence-2 using both cosince similarity and euclidean distance.

Introduction

For our reproducibility project, we chose to reproduce Disease Prediction and Early Intervention System Based on Symptom Similarity Analysis. Sentence similarity is a task the healthcare professionals perform, but it is also a well researched task in natural language processing.

Healthcare Application: Physicians perform similarity assessment when mapping a patient's symptom statement to the catalog of medical conditions. This similarity analysis is where models, like the one we will discuss, are applicable to the healthcare industry.

Ablations

Planned Ablations

- Utilize a different source containing healthcare specific information (e.g. clinical notes)
- Leverage Med2Vec instead of Word2Vec

Actual Ablations

- Introduction of additional parsing methods: ‘SpaCy’ and ‘Raw’
- Introduction of concurrency support in data pre-processing *stage*.
- Google News Word2Vec vs. MSRP Corpus based Word2Vec

Why we changed..

- Genuine interest, quality-of-life benefits (e.g., fast data processing), we gained a better understanding of original paper during implementation