Hierarchical attention networks for document similarity measurement

PhD Wei-Fan Chen
wei-fan.chen@uni-weimar.de

PhD Yamen Ajjour
yamen.ajjour@uni-weimar.de

Thang, Nguyen
thang.nguyen@uni-weimar.de

Lukas, Trautner
lukas.peter.trautner@uni-weimar.de

Anh Phuong, Le
anh.phuong.le@uni-weimar.de

Ruta Bakhda
ruta.hareshbhai.bakhda@uni.weimar.de

Abstract

Document similarity has to determine how close two pieces of text are both in surface closeness (**lexical similarity**) and meaning (**semantic similarity**). Measuring the similarity between documents is an important operation in the text processing field. In this report, we describe how a hierarchical attention network [4], which originally proposed for document classification problem, could be utilized for measuring document similarity. The model has two distinctive characteristics: it has a hierarchical structure that mirrors the hierarchical structure of document, and it has two levels of attention mechanisms applied at the word-and sentence-level. These features enable it to attend differentially to more and less important content when constructing the document representation, which is important to determine whether a pair of document are about the same side or not. We evaluated the performance of our model on three large scale political document datasets.

1 Introduction

Document similarity measurement is one of the fundamental task in Natural Language Processing. The goal is to determine whether 2 input documents have the same stance/side or not. The problem could be easily splitted out into 2 separated tasks:

- How to represent each document
- How to compare document representations

1.1 Document representation

For the first task, it is also well-known as the first step of document classification problem. Traditional approaches of document classification represent each document with sparse lexical features, such as n-grams, and then use a linear model or kernel methods on this representation to compare meaning of these documents. More recent approaches used deep learning, such as convolutional neural networks and recurrent neural networks based on long short-term memory (LSTM) to learn document representations.

Although neural-network–based approaches to document classification have been quite effective, in our implementation we test the hypothesis that better representations can be obtained by incorporating knowledge of document structure in the model architecture. The intuition underlying our model is that not all parts of a document are equally relevant for answering a query and that determining the relevant sections involves modeling the interactions of the words, not just their presence in isolation.

The Hierarchical Attention Network (HAN) we used here is designed to capture two basic insights about document structure. First, since documents have a hierarchical structure (words form sentences, sentences form a document), we likewise construct a document representation by first building representations of sentences and then aggregating those into a document representation. Second, it is observed that different words and sentences in a documents are differentially informative. Moreover, the importance of words and sentences are highly context dependent, i.e. the same word or sentence may be differentially important in different context. To include sensitivity to this fact, our model includes two levels of attention mechanisms - one at the word level and one at the sentence level - that let the model to pay more or less attention to individual words and sentences when constructing the representation of the document.

1.2 Similarity measurement

A lot of measures have been proposed for computing the similarity between two vectors. The Kullback-Leibler divergence [2] is a non-symmetric measure of the difference between the probability distributions associated with the two vectors. Euclidean distance [3] is a well-known similarity metric taken from the Euclidean geometry field. Manhattan distance [3], similar to Euclidean distance and also known as the taxicab metric, is another similarity metric. The Canberra distance metric [3] is used in situation where elements in a vector are always non-negative. Cosine similarity [1] is a measure taking the cosine of the angle between two vectors. In our implementation, the last method using cosine similarity is used, but it could be easily changed to any other above-mentioned approaches.

2 Challenges

Before we came up with the idea of using hierarchical attention network (HAN) for solving the problem of measuring the similarity between documents, we have tried to implement and run several other approaches, including using word2vec + cnn/lstm. However, no approach showed positive result.

There are some potential reasons explaining why these approaches failed. Models we have tried so far work on word-level (and this is also the most common approach for text-related problem in general and document classification in particular, beside character-level family). However, we think that for similarity classification, word-level methods have some shortcomings:

- In binary document classification problems (spam/not spam, positive/negative, etc...), assume that we have 3 documents. If document 1 and document 2 belong to different classes, and document 1 and document 3 belong to different classes, then we could conclude that documents 2 and 3 are in the same class. From ML/DL model point of view, the model could capture/memorise key words which are common for each class. For example, **beautiful, nice, interesting, amazing** should be common word in positive feedbacks, whereas **terrible, bad, silly** should be often seen in negative ones. However, for similarity classification, this ”trick” does not work. Again, assume that we have 3 documents, one in politics, one in sport, one in education. It is effortless to see that document 1 and document 2 are different, document 1 and document 3 are different, **BUT** document 2 and document 3 are different as well. **Beautiful** and **terrible** appearing in similar documents or **nice** and **interesting** appearing in opposite documents are not something rare anymore. We do not say that word-level mechanism is useless, but is it not sufficient. We admit that determining whether 2 documents/arguments are on the same side or not (our task) is easier than concluding them on the same topic or not. However, a model works for the latter should work without problem for the former. And, it also brings us possibility to broaden the application area of the model.
- Word-level model fails in case 2 documents are very similar in appearance, but actually in opposite view. If we have 2 documents:
  - Today is a beautiful day, isn’t it?

- Today isn't a beautiful day, is it?

Word-based model, including word2vec ones will undoubtedly fail to identify their opposite manners. CNN may or may not work. LSTM should deal with it better. But LSTM alone is not enough, we believe, since it is quite common that LSTM run in bi-direction, which could mess up the word **not** from different sides on the above mentioned example.

That is the reason why we think there should be something else integrated to solve the challenge. After searching for several sources, we found out an old but very interesting paper **Hierarchical attention networks for document classification**. In this paper, not only word-level but also sentence-level are utilized for classifying document. We "borrow" this idea to build my own model for solving our task.

3 Original model

Hierarchical Attention Networks consists of the following parts:

- Embedding layer.
- Word Encoder: word level bi-directional GRU to get rich representation of words.
- Word Attention: word level attention to get important information in a sentence.
- Sentence Encoder: sentence level bi-directional GRU to get rich representation of sentences.
- Sentence Attention: sentence level attention to get important sentence among sentences.
- Fully Connected layer + Softmax (This softmax layer is removed in my model. We will explain later)

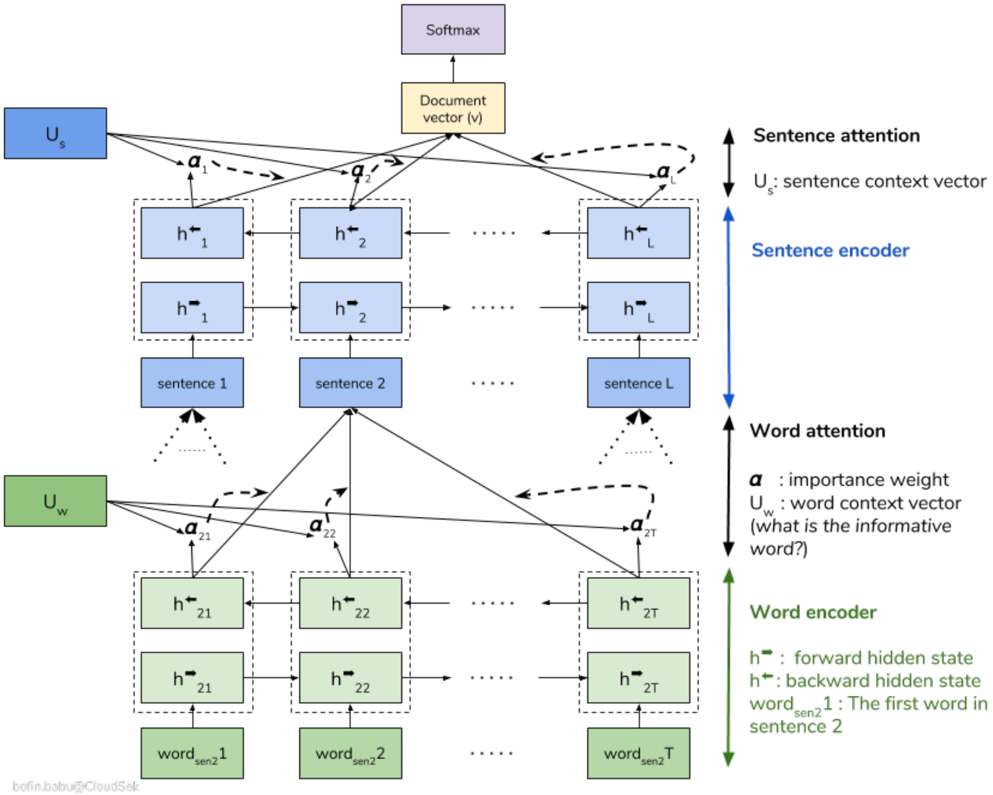

Figure 1: Hierarchical attention networks

The idea behind the model is Words make sentences and sentences make documents.The intent is to derive sentence meaning from the words and then derive the meaning of the document from those sentences. But not all words are equally important. Some of them characterize a sentence more than others. Therefore we use the attention model so that sentence vector can have more attention on “important” words. Attention model consists of two parts: Bidirectional RNN and Attention networks. While bidirectional RNN learns the meaning behind those sequence of words and returns vector corresponding to each word, Attention network gets weights corresponding to each word vector using its own shallow neural network. Then it aggregates the representation of those words to form a sentence vector i.e it calculates the weighted sum of every vector. This weighted sum embodies the whole sentence. The same procedure applies to sentence vectors so that the final vector embodies the gist of the whole document. Since it has two levels of attention model, therefore, it is called hierarchical attention networks.

4 Our model

In order to build our own model for similarity classification, based on HAN model described above, we made several modification:

- In the forward part, instead of putting a single document through the model, now we put 2 documents need to be compared through the model.
- Softmax layer at the end are removed. So it means that after each document goes through the model, what we have is not a vector of probability that document belong to each class, but a vector representing the document. The vectors corresponding to 2 documents are compared to each other based on the following formular:

  $$distance(d_1, d_2) = e^{-|x_1 - x_2|}$$

  with x_i is the output vector of document d_i we have after run the document through the model. So, if 2 vectors are similar, distance = 1, and if 2 vectors are very different from each other, distance = 0
- Instead of using Cross Entropy Loss, I used Mean Squared Error as loss, because the output of the previous step - the distance - will be a single float number within the range (0,1), and this number will be compared directly with the label 0 or 1 (In case of original model, the output for each document will be a vector of n probability, with n is the number of class, and of course in this case, cross entropy loss is the most suitable choice)
- In the original paper, it is said that the authors trained their own word2vec. In my case, we found out that using available pre-trained model provided by Stanford university (GLOVE) link or Facebook (FastText) link give my good enough performance. We picked up the 50 dimensional word2vec model from GLOVE (The lightest and most simple one) for our model

The way 2 documents are compared to each other is visualized in Figure 2

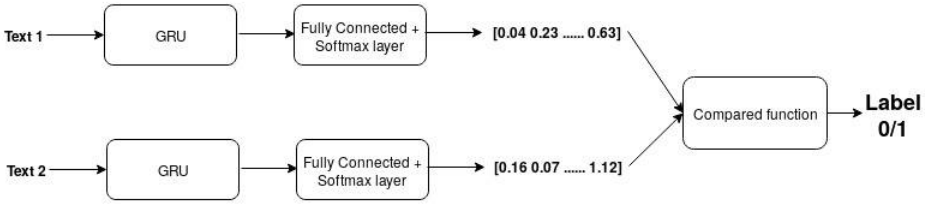

Figure 2: Documents’comparison

5 Experiments

5.1 Dataset

We evaluate the effectiveness of our model on three large scale document datasets. The statistics of the data sets aresummarized in Table 1. We use 80% of the data for training, 10% for validation, and the remaining 10% for test.

- **Same Side Stance** dataset: The dataset used in the task are derived from the following four sources: idebate.org, debatepedia.org, debatewise.org and debate.org. The training set contains arguments for a set of topics (abortion and gay marriage) and the test set contains arguments related to the same set of topics (abortion and gay marriage)
- **Same Side Political Position** dataset: This dataset is derived from the Canadian Hansard dataset. The label is based on political position of author
- **Same Side Political Stance** dataset: This dataset is also derived from the Canadian Hansard dataset. However, the label now is based on political stance instead of position of author. In other word, 2 politicians from the same party could have different opinions on a specific topic. Detail statistics is on Table 4

Table 1: Dataset statistics

	SS Stance	**SS Political Position**	**SS Political Stance**
Type of label	Stance	Political position	Political stance
Type of data	Regular	Parliamentary	Parliamentary
Number of pairs	63843	84752	45056

5.2 Result

In total, we carry out 9 experiments, including 3 within-domain experiments (training and validation sets from the same datasets) and 6 cross-domain experiments (training and validation sets from different datasets). Graph for training (**lower figure**) and validation (**upper figure**) curves for each experiments will be shown in the following section.

5.2.1 Training and validation statistics

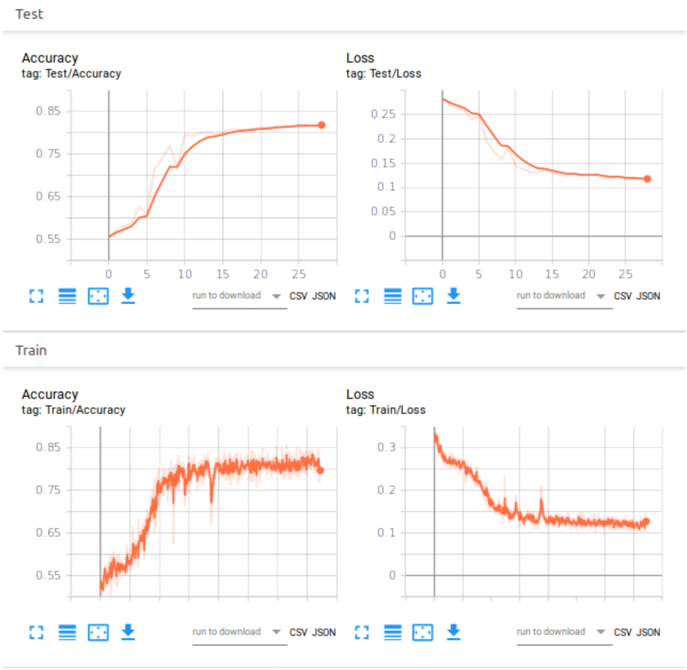

Figure 3: Training and validation sets from Same Side Stance

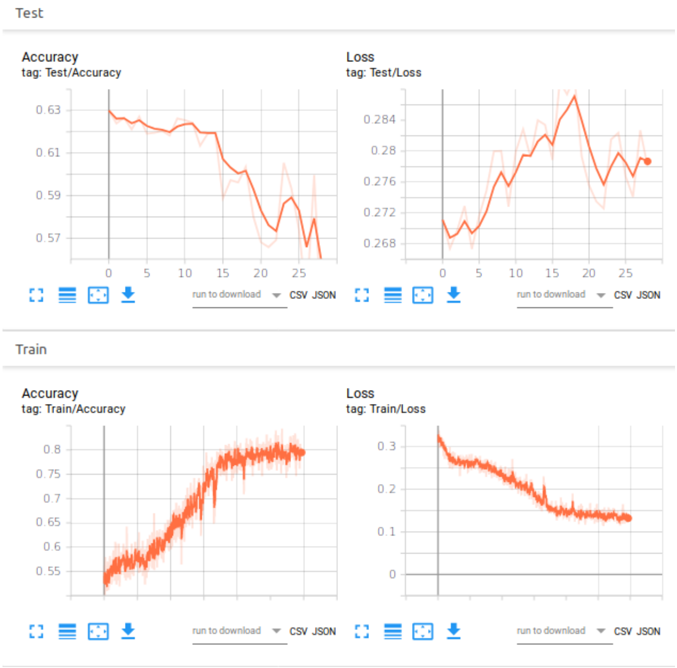

Figure 4: Training set from Same Side Stance and validation set from Same Side political position

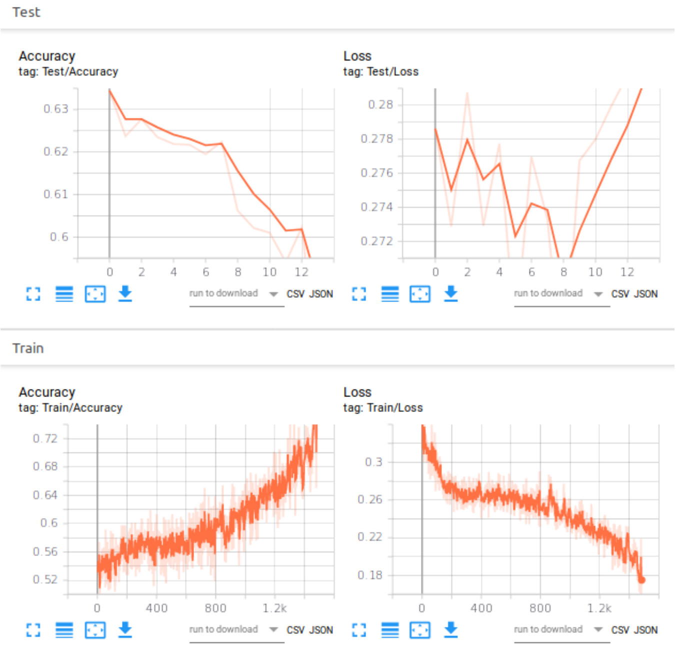

Figure 5: Training set from Same Side Stance and validation set from Same Side political stance

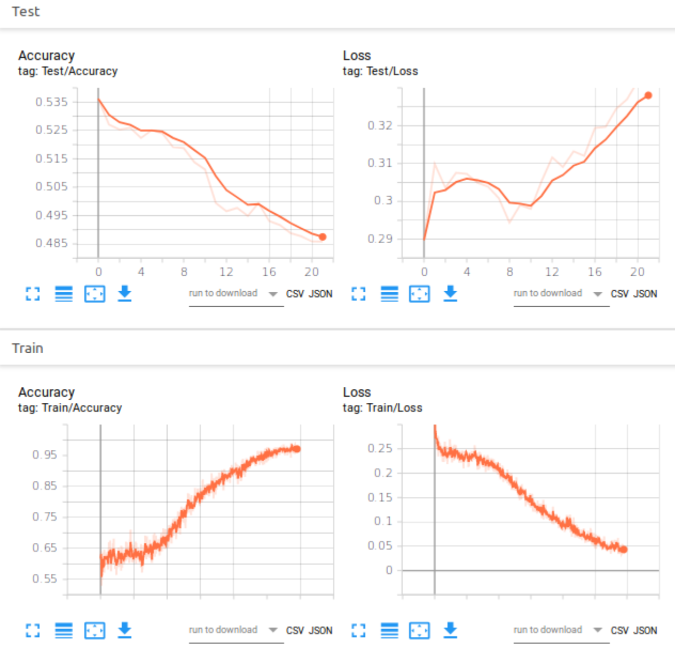

Figure 6: Training set from Same Side political position and validation set from Same Side Stance

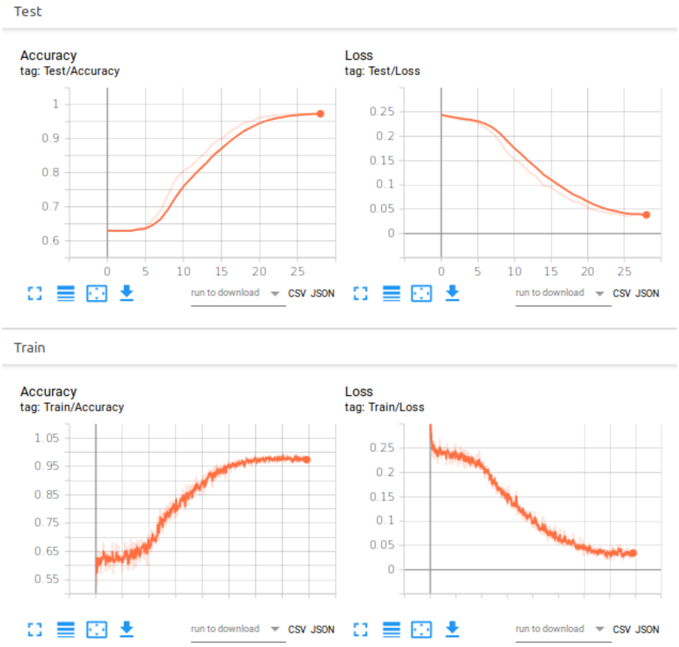

Figure 7: Training and validation sets from Same Side political position

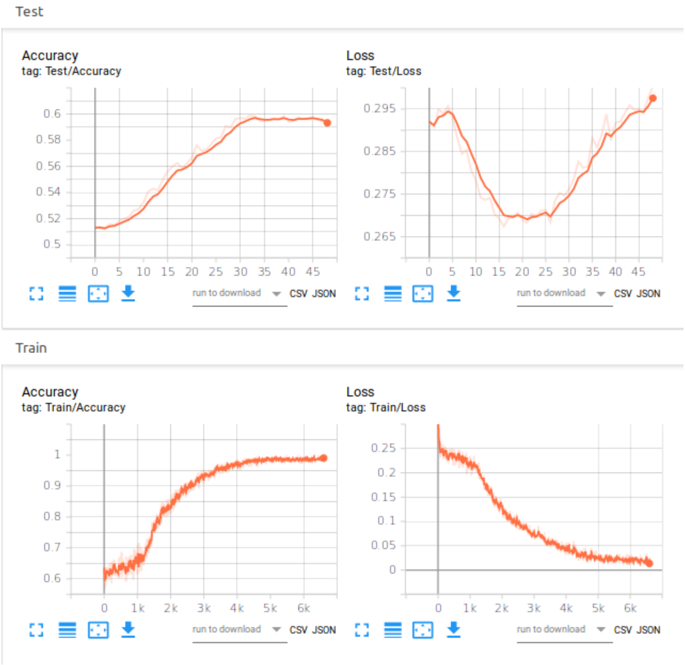

Figure 8: Training set from Same Side political position and validation set from Same Side political stance

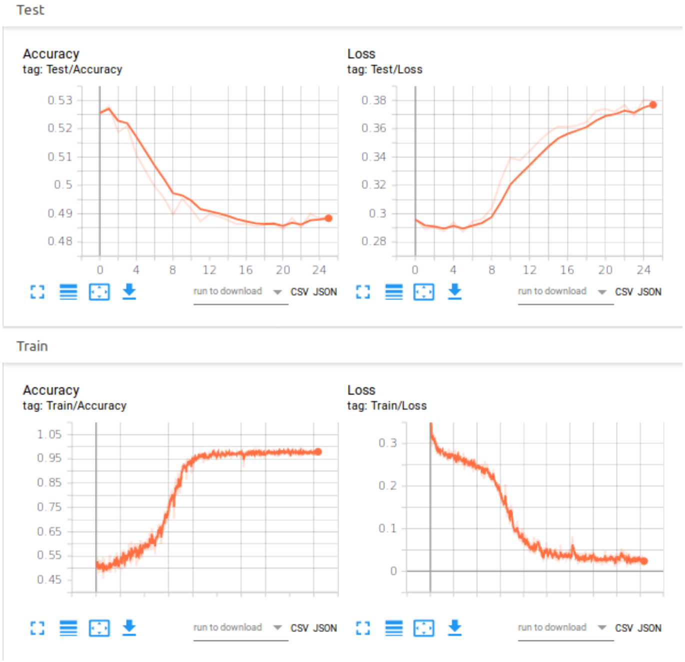

Figure 9: Training set from Same Side political stance and validation set from Same Side Stance

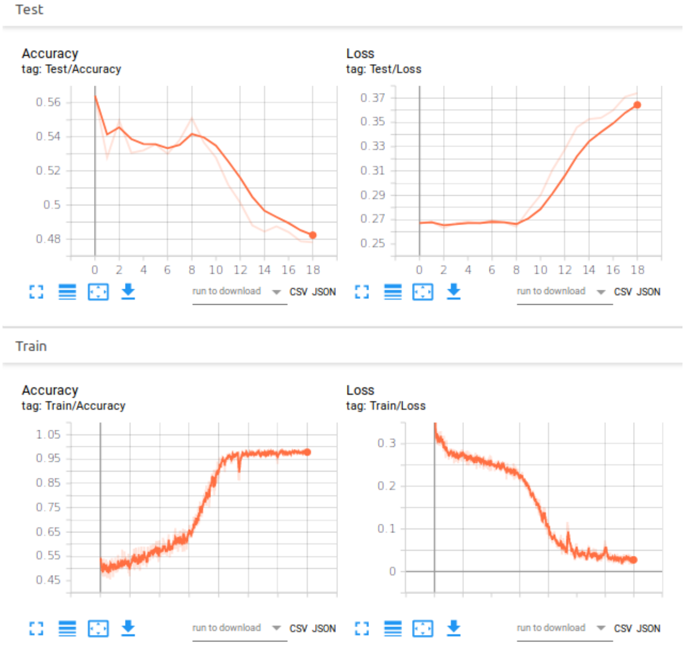

Figure 10: Training set from Same Side political stance and validation set from Same Side political position

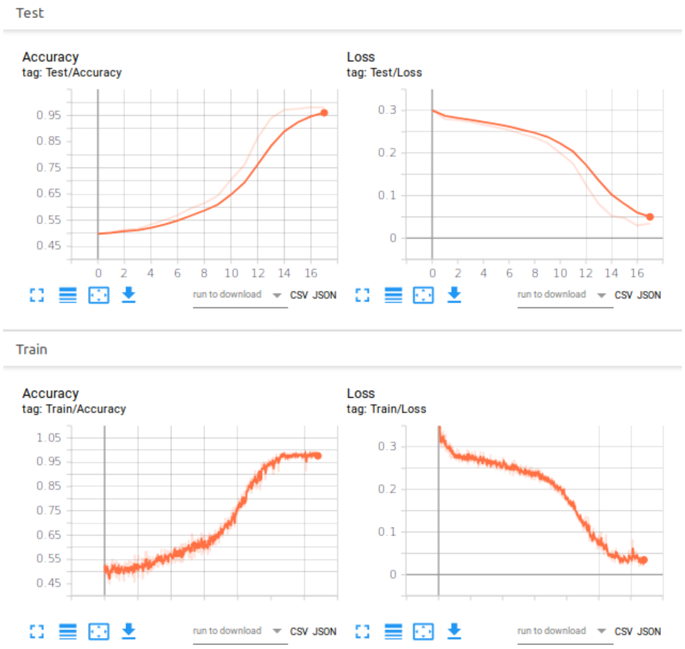

Figure 11: Training and validation sets from Same Side political stance

5.2.2 Test statistics

Table 2 shows general accuracy for each experiment. Table 4 shows detail test's statistics for each experiment.

Table 2: Test set accuracy (row for training set, column for test set)

	SS Stance	SS Political Position	SS Political Stance
SS Stance	79.83%	52.63%	57.70%
SS Political Position	48.8%	95.16%	57.12%
SS Political Stance	47.20%	45.65%	94.92%

Table 3: Number of pairs for each class

	Same side	Different side	Total
SS Stance	6822 (53.38%)	5959 (46.62%)	12781
SS Political Position	10767 (63.52%)	6184 (36.48%)	16951
SS Political Stance	4642 (51.72%)	4333 (48.28%)	8975

- **Same Side stance**: 1.
- **Same Side political position**: 2
- **Same Side political stance**: 3

Table 4: Detail statistics for test sets

Trained on 1, tested on 1	**Precision**	**Recall**	**F1 score**
Same side	85%	80%	82%
Different side	78%	83%	81%
Trained on 1, tested on 2	**Precision**	**Recall**	**F1 score**
Same side	63%	98%	77%
Different side	31%	02%	03%
Trained on 1, tested on 3	**Precision**	**Recall**	**F1 score**
Same side	52%	89%	66%
Different side	49%	11%	18%
Trained on 2, tested on 1	**Precision**	**Recall**	**F1 score**
Same side	54%	80%	65%
Different side	51%	23%	32%
Trained on 2, tested on 2	**Precision**	**Recall**	**F1 score**
Same side	99%	97%	98%
Different side	94%	99%	97%
Trained on 2, tested on 3	**Precision**	**Recall**	**F1 score**
Same side	56%	71%	63%
Different side	56%	40%	47%
Trained on 3, tested on 1	**Precision**	**Recall**	**F1 score**
Same side	53%	57%	55%
Different side	47%	43%	45%
Trained on 3, tested on 2	**Precision**	**Recall**	**F1 score**
Same side	63%	70%	66%
Different side	36%	29%	32%
Trained on 3, tested on 3	**Precision**	**Recall**	**F1 score**
Same side	99%	97%	98%
Different side	96%	99%	98%

6 Explanation

Amongst 9 experiments, as mentioned aboved, we could group them into 2 subsets: Within-domain with 3 experiments and cross-domain with 6 experiments. In the following section, we will analyze results for both.

6.1 Within-domain experiments

As we could see from Table 2, accuracy for test set in all 3 experiments are quite promising. These figures demonstrate that our model is capable to capture the relation amongst documents in normal cases.

6.2 Cross-domain experiments

In these experiments, we apply transfer learning with the expectation that our model could apply what it learns from one dataset to predict the relation between pairs of documents from other dataset. In general, accuracy for test set is very low (<60%). After analyzing the stucture and the source of each dataset, we find out some potential reasons for this low performance:

- **Difference in datasets' area**: The first dataset (Same Side Stance) only contains arguments for 2 topics (abortion and gay marriage). On the other hand, the other 2 datasets are composed of arguments from various topics, including the 2 above-mentioned topics. As a result, models trained on the first dataset are lack of knowledge about topics like education or sport when predicting data from the second or the third datasets. Vice versa, Models trained on the last 2 datasets are "diluted" with redundant topics when predicting data from the first dataset.
- **Difference in datasets' speaker's occupation**: The first dataset contains arguments from people with different occupations, while only politicians' arguments are collected for the second and the third datasets. As a result, the tongue of the first dataset seems to be less formal, in compared to the others.
- **The gap between party side and opinion**: Even the Same Side political position and Same Side political stance dataset are from the same sources, models trained on one and validated/tested on the other still produce an under-performance. It mainly comes from the point that 2 politicians from the same party do not necessarily have the same opinion about a specific topic. As a result, a model purely trained on one and validated/tested on the other is likely to fail to predict document relation correctly.

7 Contribution

There are 2 main contribution of this model to the whole project:

7.1 Document relation estimation

All other models have been used in this project are used for analyzing a single document independently. However, This hierarchical attention network has the capability to estimate the relation between documents. Hence, this model has a clear advantage: Bring us an overview about the connection between document's content and the similarity/difference in stance of author's document.

7.2 Cross measurement

The second contribution, which is even more important than the first one, is cross measurement. One crucial task of **Visit The Dome** is to classify given a document, whether the author's stance is positive or negative. As mentioned above, all other models have been tried so far only evaluate the document based on its own content, which is possibly biased. however, our hierarchical attention network could be used to evaluate the relation between this documents and many other documents, before summarizing predictions into final decision.

For example, assume we have 20 labelled documents belonging to the same topic, 10 of them are positive, the others are negative. Given a new document, we could determine the label for it by doing the following steps:

- **Step 1**: Pair it to each of 20 above-mentioned documents.
- **Step 2**: Use each pair as one input for our hierarchical attention network.
- **Step 3**: Combine 20 predictions. If 7/10 predictions for this document and positive ones are positive, and only 5/10 prediction for this document and negative ones are negative, then we could conclude that this document's label is positive. Certainly we could define more sophisticated logic for decision-making.

8 Future improvements

There are rooms for improvement, includes:

- Use a higher dimensional word2vec model (100, 200 or 300 dimension)
- Use smooth l1 loss instead of MSE, since the former is well-known for dealing well with outlier
- Use different approach to create training set. At the moment is the similarity score $\geq$ 0.5, we say 2 documents are from the same side/stance. Otherwise they are different. However, it is not convincing enough even for human to say 2 scores 0.49 and 0.51 are really different. Hence, it makes sense if we ignore pair of documents whose score is within a pre-defined range, for example $[0.45, 0.55]$
- Try other optimizers, like Adam or Rmsprop. Additionally, instead of using default learning rate schedule, I would like to try more complex one. For example, as suggested in [5], learning rate could be halved every 3 epoches for 10 times.
- Based on well-known word2vec model, we could further train our own embedding layer, which focus more on set of words used frequently on our dataset.
- Fine-tune parameters. The current set of parameters is not the best one. 30 epochs is what I set at the beginning, but the test loss was still decreasing.

References

[1] J. Han, J. Pei, and M. Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[2] S. Kullback and R. A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[3] T. W. Schoenharl and G. Madey. Evaluation of measurement techniques for the validation of agent-based simulations against streaming data. In *International Conference on Computational Science*, pages 6–15. Springer, 2008.

[4] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489, 2016.

[5] X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.