MLAF: Multi-Level Attention Fused Model for Sentence Similarity Measurement

Mengya Chen¹, * and Xiaoqi Sun²

¹School of Computer Engineering and Science, Shanghai University, Shanghai, China
²School of Computer Engineering and Science, Shanghai University, Shanghai, China
*Corresponding author: ashleychen@shu.edu.cn

Abstract. Semantic textual similarity measurement is an essential task in text mining and natural language processing, which has profound research value. Recent LSTM-based methods have achieved attractive success by modelling sentences as fixed-length vectors. However, since the LSTM network can only extract the semantic information of sentences sequentially, the long-term dependency of sentences cannot be efficiently captured when the sequence length is long. Moreover, current models treat each syntactic component in the sentence equally, while the effects of different syntactic elements to the sentence meaning expression are different. Generally, nouns, verbs, and adjectives contribute more than prepositions and conjunctions to the semantic meaning of the sentence. In this paper, we propose a multi-level attention fused (MLAF) model, which correlates self-attention and word attention with sentence representations, extracting the semantic information of sentences from different levels. The self-attention supplements sequential operations to model long-term dependency of sentences. The word attention built upon raw sentence highlights the vital syntactic components of sentences. Experiments on different datasets demonstrate that the proposed MLAF model has achieved significant improvements over baselines.

1. Introduction

Measuring sentence similarity aims to predict the relatedness of the sentence pair, and it plays an irreplaceable role in many text mining and natural language processing tasks. Information retrieval problems such as text classification and text clustering and other NLP tasks such as machine translation and text summarization are in support of sentence similarity methods [1]. Although researchers have proposed many models to measure the similarity of sentences, it is still challenging due to the diversity of linguistic expression and the limited training data [2].

Owing to the success of word embedding [3][4], researchers have attempted to measure sentence similarity by using sentence embedding models based on neural networks [5]. Despite the success they made, there are still two problems.

The first problem is that the long-range context dependency of sentences can not be modeled adequately by these methods which adapt a long short-term memory (LSTM) network when the sentence length is too long. Compared with words, sentences with the same meaning have a different structure, and it is difficult to map the sentence as a fixed-length vector. The second problem with current sentence similarity methods is that these models treat each syntactic component in sentences equally, which is different from the way that humans read and understand sentences. Humans will pay more attention to soma words and read other words superficially, especially when they try to understand long texts. In a
nutshell, different syntactic components in a sentence have different effects on the understanding of sentence semantic meaning [6].

The observations above motivate us to develop a model that can extract the long-term dependency of sentences effectively and selectively give more weights to valuable content. Concretely, we use a multi-level attention fused model (MLAF) to achieve a good sentence representation. The LSTM language model encodes the semantic information of the sentence and the subsequent self-attention layer is used to improve the ability to model long-range context dependency. The network first models long-term context dependency of sentences sequentially and extracts high-level semantic information. Then, a word-level attention mechanism is employed to simulate the habit of human reading sentences. The word attention supplements previous sequential sentences representations with low-level word representations, extracting semantic information of sentences at a different abstractive level without any complicated sequential operations. Finally, the results of the two parts are fused to obtain the final representations of sentences. To summarize, the main contributions of this paper are:

- We present a new sentence representation model. Our model uses two integrated attention mechanisms, self-attention aims to extract long-range dependency, while word attention aims to give more weights to important syntactic components.
- We model sentences from different abstractive levels. The high-level sequential information is tightly fused with the low-level information of the raw sentence.

![Figure 1. The overall architecture of the proposed multi-level attention fused model, including encoding layer, self-attention layer, word level attention layer, attention fusion layer, and similarity measurement layer.](image)

2. General Siamese Neural Model for Sentence Similarity Measurement

Given a sentence pair \( S^i(x_1, x_2, \ldots, x_n) \) and \( S^r(x_1, x_2, \ldots, x_m) \), the sentence similarity measurement is usually regarded as a classification or regression task, depending on the dataset used. Recently, the LSTM networks have been widely applied in this task because of their natural advantages in processing sequential information. [7] proposed a Siamese LSTM model (MaLSTM) for calculating sentence similarity, which can be seen as the general framework of this task. There are two LSTM networks \( LSTM^i \) and \( LSTM^r \) that share weights in MaLSTM, and each network process one sentence in pair.
2.1. Encoding Layer

The sequences of tokens \( (x_1, x_2, \ldots, x_n) \) and \( (x_1, x_2, \ldots, x_m) \) are firstly converted to sequences of word vectors \( (x_{v,1}, x_{v,2}, \ldots, x_{v,n}) \) and \( (x_{v,1}, x_{v,2}, \ldots, x_{v,m}) \). Then, the representation is passed to the LSTM network, learning a mapping from the word vectors to a fixed-length semantic representation. Each token in sentences can be represented as:

\[
    h^l_t = \text{LSTM}^l(x_{v,t}, h^l_{t-1}) \\
    h^r_t = \text{LSTM}^l(x_{v,t}, h^r_{t-1})
\]

As a result, the representations of sentences are \( H^l = (h^l_1, h^l_2, \ldots, h^l_n) \) and \( H^r = (h^r_1, h^r_2, \ldots, h^r_m) \).

2.2. Similarity Measurement Layer

To force the sentence model completely capture the complex semantic information of the sentence, rather than supplementing the network with other learner, MaLSTM uses a simple similarity function \( f(h^l_n, h^r_m) \). It can be represented as:

\[
    f(h^l_n, h^r_m) = \exp \left( -\|h^l_n - h^r_m\| \right) \in [0,1]
\]

3. Multi-level Attention Fused Sentence Similarity Model

In this section, we will introduce the MLAF sentence similarity model. A good sentence representation method can not only effectively learn the long-range dependency of a sentence even when the sentence length is long, but also can selectively give more weights to important syntactic parts in sentences. We designed the model with this idea in mind, exploiting the multi-level attention fused network to model sentence from different aspects. The overall framework of our model is illustrated in figure 1.

3.1. Encoding Layer

For the sentence pair \( S^l(x_1, x_2, \ldots, x_n) \) and \( S^r(x_1, x_2, \ldots, x_m) \), each word \( x_t \) in sentences is mapping into a vector \( x_t \) by looking up the word embedding matrix \( W_{\text{word}} \in \mathbb{R}^{d \times |v|} \), where \( d \) is the size of word embedding, and \( |v| \) is the number of vocabulary. \( W_{\text{word}} \) is initialized by pre-trained word embedding. Furthermore, we also utilize the Part-of-Speech (POS) tags as the input to the model. Assign a serial number to each type of tags and map it to a vector \( c_{v,t} \in \mathbb{R}^d \), where \( d \) is the embedding size and the vector is initialized by normal distribution. And then, the word embedding vector \( x_{v,t} \) and POS embedding vector \( c_{v,t} \) are concatenated as the representation of word \( x_t \). Finally, we apply a LSTE network to encode each token, getting the semantic representation \( h_t \).

3.2. Word Attention Layer

In order to assign more weights to the essential syntactic contents in the sentences, and to make the sentences representations can be directly connected with more word-level information of raw sentences without any complicated sequential calculations, we supplement a word-level attention mechanism to our model.

Typical attention mechanism models the dependency between hidden states of LSTMs and sentence representations and summarizes such relationship into context vectors. While our approach is using Part-of-Speech (POS) embedding, calculating attention weights directly on the input sequence. Concretely, we calculate the dot product of each word embedding vector and corresponding POS embedding vector, which is described as:

\[
    \text{att}(x^l_t) = \frac{\exp (x^l_{v,t}, c^l_{v,t})}{\sum_{t=1}^{n} \exp (x^l_{v,t}, x^l_{v,t})}, \text{att}(x^r_t) = \frac{\exp (x^r_{v,t}, c^r_{v,t})}{\sum_{t=1}^{m} \exp (x^r_{v,t}, x^r_{v,t})}
\]

Where \( \exp \) refers to the exponential function, \( \text{att}(x^l_t) \), \( \text{att}(x^r_t) \) are the weights that determines the relative power of each token with different POS tag. Then we use weighted summation of word embedding vectors to incorporate the attention weights into sentence presentations:
\( S_{\text{att}}^l = \frac{1}{n} \sum_{i=1}^{n} \text{att}(x_i^i)x_{i,j}^l, S_{\text{att}}^r = \frac{1}{m} \sum_{j=1}^{m} \text{att}(x_i^i)x_{i,j}^r \)  

(5)

where \( S_{\text{att}}^l \) and \( S_{\text{att}}^r \) represent the word-level representation of the sentences.

### 3.3. Self-Attention Layer

The LSTM language models typically learn semantic information of sentences along the word positions of the input sequences which can capture the long-term context dependency of sentences to some extent. However, it is precisely because of this inherently sequential nature that precludes the ability of LSTMs to model long-range dependency when sentence length is longer. In order to solve this problem, we add as self-attention layer after the LSTM layer.

The attention mechanism used in natural language processing can essentially be described as a mapping from a query to a series of key-value pairs, And in self-attention, query = key = value. For example, given a sentence, then each word will calculate with all words in sentences directly. The purpose is to learn to token dependency within the sentence, capturing the internal structure of the sentence.

Formally, for sentences \( S^l(x_1, x_2, ..., x_n) \) and \( S^r(x_1, x_2, ..., x_m) \), we first compute the self-attention matrix:

\[
A_{i,j}^l = \langle h_i^l, h_j^l \rangle, A_{i,j}^r = \langle h_i^r, h_j^r \rangle
\]

(6)

where \( A_{i,j}^l \) and \( A_{i,j}^r \) represent the relevance between \( i \)-th word and \( j \)-th word in sentences. \( h_i^l, h_j^l, h_i^r, h_j^r \) are corresponding LSTM outputs. \( \langle.,. \rangle \) represents the different similarity function, and in this paper, we use dot product. Then we calculate the self-attention weights of each word by the following formula:

\[
s_i^l = \text{softmax}(A_{i,i}^l), s_i^r = \text{softmax}(A_{i,i}^r)
\]

(7)

To capture all the information and highlight the important properties of the two sentences, we perform a mean pooling and a maximum pooling on each sentence, and then we apply a concatenate operation on them:

\[
S_{\text{mean}}^l = \frac{1}{n} \sum_{i=1}^{n} s_i^l, S_{\text{max}}^l = \max_{i=1,...,n} s_i^l
\]

(8)

\[
S_{\text{mean}}^r = \frac{1}{m} \sum_{j=1}^{m} s_j^r, S_{\text{max}}^r = \max_{j=1,...,m} s_j^r
\]

(9)

\[
S^l = [S_{\text{mean}}^l \oplus S_{\text{max}}^l], S^r = [S_{\text{mean}}^r \oplus S_{\text{max}}^r]
\]

(10)

where \( \oplus \) is concatenation operation. Due to each word is directly calculated with all words in the sentence, long-term dependency can be captured even when the sentence length is long.

### 3.4. Fusion Layer for Word Attention and Self-Attention

In this part, we will introduce the fusion layer to get the final representations of the sentences. The fusion representations \( V^l, V^r \) will be sent to the similarity measurement layer to calculate the similarity grades.

\[
\hat{h}^l = [S_{\text{mean}}^l \oplus S_{\text{max}}^l \oplus S_{\text{att}}^l], \hat{h}^r = [S_{\text{mean}}^r \oplus S_{\text{max}}^r \oplus S_{\text{att}}^r]
\]

(11)

\[
V^l = \text{Relu}(W_h^l \hat{h}^l + b_h^l), V^r = \text{Relu}(W_h^r \hat{h}^r + b_h^r)
\]

(12)

where \( \hat{h}^l, \hat{h}^r \) are the concatenation of self-attention and word attention, \( V^l, V^r \) are the sentence embeddings. \( W_h^l, b_h^l, W_h^r, b_h^r \) are the weights and biases matrix of corresponding sentences.

### 4. Experiments and results

In this section, we will introduce the datasets, the experimental settings, and the results of comparison with other methods.
4.1. Datasets and Experimental Settings
We use 8 datasets from Semantic Textual Similarity (STS) task, which covers a wide range of domains. In our model, word embedding and POS embedding are 300 dimensions, which are initialized respectively by pre-trained Glove vectors and normal distribution with $\mu = 0.0, \sigma = 0.01$. And the word vectors and POS vectors are finetuned during the procedure of training. The dimension of LSTM hidden units is 50, and it is initialized by 0. The mini-batch size is 32. We use the MSE loss function to train our model. We apply the Adam for training, and the learning rate is set to 0.01. The epoch is 100. We also use a dropout strategy to avoid over-fitting.

4.2. Overall results
We use the Pearson correlation coefficient that is widely used to measure the degree of correlation between two variables to evaluate the performance of MLAF. The testing results of our approach and several prior works are summarized in Table 1, compares to baselines, our MLAF sentence similarity model achieves great improvements on most datasets. These results directly demonstrate the competitiveness and effectiveness of our sentence similarity model. Besides, we reckon that learn the reason that why our model does not achieve the best results on some datasets can help us conduct further research.

Going back to OnWN-2012, and OnWN-2013 datasets, we find that the datasets consist of phrases rather than sentences, and most phrases are no longer than ten words. We hypothesize that in order to force our sentence similarity model to capture long-term dependency on long sentences, we set the max length of the sentences to 25, which introduces noise on these datasets.

In the SMTeuroparl-2012 and SMTnews-2012 datasets, there are a large number of special characters, digital information, and polysemous words, such as (H-0886/00), 5.2%. Moreover, these particular properties rarely exist in our training set. From this phenomenon, we can know that our MLAF model lacks the keen ability to handle this particular information very well.

| Datasets          | Compared methods |
|-------------------|------------------|
|                  | [6]  | [8]  | [4]  | [9]  | [7]  | [5]  | [6]  | MLAF |
| MSRpar            | 0.465 | 0.168 | 0.469 | 0.448 | 0.428 | 0.455 | 0.497 | 0.504 |
| MSRvid            | 0.792 | 0.417 | 0.625 | 0.452 | 0.752 | 0.722 | 0.846 | 0.848 |
| SMTeuroparl       | 0.521 | 0.352 | 0.421 | 0.447 | 0.413 | 0.402 | 0.493 | 0.487 |
| OnWN              | 0.725 | 0.297 | 0.536 | 0.628 | 0.563 | 0.331 | 0.725 | 0.615 |
| SMTnews           | 0.652 | 0.308 | 0.567 | 0.376 | 0.514 | 0.600 | 0.664 | 0.617 |
| 2012 average      | 0.631 | 0.308 | 0.524 | 0.470 | 0.534 | 0.502 | 0.645 | 0.614 |
| Headlines         | 0.740 | 0.352 | 0.645 | 0.652 | 0.713 | 0.623 | 0.737 | 0.762 |
| FNWN              | 0.503 | 0.301 | 0.301 | 0.232 | 0.511 | 0.471 | 0.498 | 0.547 |
| OnWN              | 0.747 | 0.144 | 0.493 | 0.498 | 0.697 | 0.326 | 0.779 | 0.737 |
| 2013 average      | 0.663 | 0.266 | 0.480 | 0.461 | 0.640 | 0.473 | 0.671 | 0.682 |

5. Conclusion
In this paper, we propose a multi-level attention fused sentence model to calculate sentence relatedness. Our model extracts semantic information of sentences from different levels through self-attention mechanism and word attention mechanism. Since each word can be directly calculated with all the words in the sentence, the self-attention mechanism can supplement LSTM, capturing the long-distance dependency of the sentence. Word attention mechanism can directly calculate the attention weights at the source side without sequential operation and give more weights to important syntactic contents. The experimental results show that the proposed model can effectively improve the performance of sentence similarity measurement, extracting long-term dependency of sentences, and selectively assign different weights to different syntactic contents.
References

[1] Pilehvar M T andNavigli R 2015 From senses to texts: An all-in-one graph-based approach for measuring semantic similarity Artif. Intell. 228 95–128

[2] H.Gomaa W and A. Fahmy A 2013 A Survey of Text Similarity Approaches Int. J. Comput. Appl. 68 13–8

[3] Mikolov T, Sutskever I, Chen K, Corrado G and Dean J 2013 Distributed Representations of Words and Phrases and their Compositionality ArXiv1310.4546 Cs Stat

[4] Pennington J, Socher R and Manning C 2014 Glove: Global Vectors for Word Representation Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (Doha, Qatar: Association for Computational Linguistics) pp 1532–43

[5] He H, Gimpel K and Lin J 2015 Multi-Perspective Sentence Similarity Modeling with Convolutional Neural Networks Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (Lisbon, Portugal: Association for Computational Linguistics) pp 1576–86

[6] Wang S, Zhang J and Zong C 2017 Learning Sentence Representation with Guidance of Human Attention Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence Twenty-Sixth International Joint Conference on Artificial Intelligence (Melbourne, Australia: International Joint Conferences on Artificial Intelligence Organization) pp 4137–43

[7] Thyagarajan A 2015 Siamese Recurrent Architectures for Learning Sentence Similarity

[8] Kiros R, Zhu Y, Salakhutdinov R, Zemel R S, Torralba A, Urtasun R and Fidler S 2015 Skip-Thought Vectors ArXiv1506.06726 Cs

[9] Kenter T, Borisov A and de Rijke M 2016 Siamese CBOW: Optimizing Word Embeddings for Sentence Representations ArXiv1606.04640 Cs