The Thirty-Third International
FLAIRS Conference (FLAIRS-33)

Attention Based Transformer for Student Answers Assessment

Nisrine Ait Khayi, Vasile Rus
Institute for Intelligent Systems and the University of Memphis, Memphis TN
ntkhynyn@memphis.edu, vrus@memphis.edu

Abstract

Inspired by Vaswani's transformer, we propose in this paper an attention-based transformer neural network with a multi-head attention mechanism for the task of student answer assessment. Results show the competitiveness of our proposed model. A highest accuracy of 71.5% was achieved when using ELMo embeddings, 10 heads of attention, and 2 layers. This is very competitive and rivals the highest accuracy achieved by a previously proposed BI-GRU-Capsnet deep network (72.5%) on the same dataset. The main advantages of using transformers over BI-GRU-Capsnet is reducing the training time and giving more space for parallelization.

Introduction

Automatically assessing open-ended, short student responses plays a vital role in the effectiveness of dialogue-based intelligent tutoring systems (ITSs) (Rus et al. 2013) and other education technologies that rely on freely generated (open ended) student input. Assessment is vital because it provides an insight about the mastery level of the student on the target topic which in turn enables the tutoring system to trigger appropriate micro-adaptation (step-level), e.g., in the form of appropriate hints, as well as macro-level adaptation, e.g., selecting appropriate instructional tasks, for struggling learners.

Assessing open-ended, short answers that students provide during their interactions with ITSs is challenging as students express their answers in a variety of ways based on their knowledge level, cognitive abilities, and other factors. This diversity of answers creates major challenges to semantic similarity approaches which are widely used to assess freely generated student responses (Banjade et al. 2016; Maharjan et al. 2018; Rus et al. 2013). Typically, such approaches work well when the two texts being compared, e.g., the student answer and the reference/expert-generated answer, are self-contained, i.e., they do not rely much on prior/wider contexts such as the previous dialogue history in dialogue-based ITSs or the problem description, e.g., a Physics problem. However, naturally occurring student responses do rely heavily on context through linguistic phenomena such as ellipsis or anaphora (Banjade et al. 2016).

Semantic similarity is a well-defined research problem in Natural Language Processing (NLP). It is a key challenge in numerous applications such as text summarization (Nenkova et al. 2011), assessing the correctness of students answers – our task in this paper (Banjade et al. 2016; Dikli 2006; Maharjan et al. 2018), question answering (Vo et al. 2015) and machine translation (Corley and Mihalcea 2005). At its core, the task of semantic similarity is about estimating the degree of similarity between two short texts, e.g., a student generated answer and an expert-generated, reference answer. It is a challenging task due to the complexity and variability of natural language statements as exemplified in Table 1 by the diversity of student responses.

Problem description:
While speeding up, a large truck pushes a small compact car.
Tutor question:
How do the magnitudes of forces they exert on each other compare?
Reference answer:
The forces from the truck and car are equal and opposite.
Student answers:
A1. *The magnitudes of the forces are equal and opposite to each other due to Newton's third law of motion.*
A2. *they are equal and opposite in direction*
A3. *equal and opposite*
A4. *the truck applies an equal and opposite force to the car.*

Table 1. Examples of student generated short answers during tutorial dialogues

To tackle the difficult and critical task of assessing student responses, we adopt here a deep learning approach. Deep learning networks have displayed superior performance on various NLP tasks, particularly text classification and semantic similarity. Motivated by this, we propose a transformer-based network for students answers assessment. The main advantages of using transformers over other approaches, e.g., Bi-GRU-Capsnet (Ait Khayi and Rus 2019), is reducing the training time and easing parallelization across elements of the input sequence. Our transformer network is composed of several encoder layers where self-attention is a major component. More specifically, our proposed model uses a multi-head attention mechanism that allows modeling dependencies regardless of their distance in the input sentence which could be either a student answer or a reference answer. This mechanism consists of several self-attention layers that run in parallel and then their outputs are concatenated. Further details are provided in a later section in the paper.

We experimented with the proposed deep neural network on the DT-Grade dataset (Banjade et al. 2016) which contains 900 instances categorized in four classes: correct (367 instances), incorrect (238 instances), correct but incomplete (210 instances), and contradictory (84 instances). To overcome the problem of class size imbalance in the dataset and given its relative small size, we consider a binary classification where all instances in the incorrect, correct but incomplete, and contradictory categories are deemed as incorrect.

The rest of the paper is organized as follows: Section 2 presents a brief review of several research works that used a transformer network for the semantic similarity task. Section 3 explains the proposed model's architecture. Section 4 summarizes the experiments we conducted to validate the effectiveness of the proposed model and provide results on the DT-Grade dataset. We end the paper with conclusions and highlighting future research directions.

Related Work

Several NLP researchers have applied attention-based mechanisms to boost the capabilities of Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) models in terms of performance and training cost when applied to text processing tasks. To this end, Vaswani and colleagues (2017) proposed a transformer network without recurrence and convolutions and based only on attention mechanisms to capture global relations between input and output texts. This transformer network allows for significantly more parallelization and yielded new state of the art results in machine translation in shorter training time. In this work, we use only the encoder component of the Transformer to assess the correctness of the students answers in dialogue based ITSs. Additional research studies have applied various encoders for the task of Short Textual Similarity (STS). Tang et al. (2018) proposed a shared sentence encoder to improve the multilingual semantic textual similarity (STS) in low resource languages with insufficient labelling (e.g. Spanish, Arabic, Thai etc..). By exploiting the nature of a multilingual encoder, one sentence can have multiple representations for different target languages which led to improved semantic similarity (between two short texts) results. Their proposed encoder STS model architecture consists of the following components: 1) Word embedding, 2) Masked Multi Self-Attention and 3) Feed Forward network. This transformer-based model architecture is different from our proposed model in numerous ways. First, only Fasttext embeddings (Bojanowski et al. 2017**)** were used in their experiments. In our work, we used three different word embeddings: Glove, ELMo and Word2vec. Second, their attention mechanism is different from ours as it is based on an inter-sentence attention that follows the approach described in (Wang et al. 2016). Our self-attention mechanism is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence. Finally, their proposed model was evaluated on a different task than ours. Our model is the first encoder transformer-based model applied for assessing the correctness of student answers in the context of dialogue-based intelligent tutoring systems. Yang et al. (2018) proposed an encoder-based network applied on the Semantic Textual Similarity (STS) benchmark and SemEval 2017's Community Question Answering (CQA) question similarity subtask. An encoder-transformer, as described in Vaswani et al. (2017), has been used to compute a sentence embedding of the input u and the response embeddings v which are passed through an additional fully connected layer to get the output v'. The final dot product between u and v' is computed to get the semantic score between the input and the response. The results of the conducted experiments for the STS Benchmark showed the competitiveness of the sentence encoding based models.

The Proposed Attention Based Model

The proposed model consists of the following components: 1) an embedding layer, 2) a positional encoding layer, 3) a transformer layer, and 4) a SoftMax layer/classifier (see Figure 1).

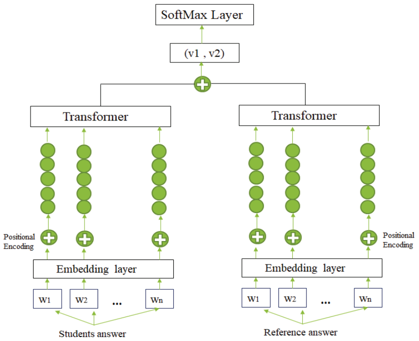

Figure 1. Model Architecture

We consider an extended student answer as the first input consisting of the concatenation of the corresponding problem description, the previous tutor question (which accounts as prior context), and the student answer. In the student answers described in Table 1, the extended student answer will be $[PD, TQ, A1]$. The inputs to the embedding layers are tokenized.

Embedding

Given a student answer X and a reference answer X', we tokenize them into a sequence of word tokens: $X = [w_1, \dots, w_n]$ and $X' = [w'_1, \dots, w'_m]$. Afterwards, each token is converted into a d-dimensional (d=300,1024) vector through the embedding layer. We considered the following three-word embeddings: Glove (Pennington et al. 2014), Word2vec (Mikolov et al. 2013), and ELMo (Peters et al. 2018).

Positional Encoding

Positional Encoding is used to capture the order of the tokens in the input. Since the embedding layer captures the meaning of words and there is no recurrence and convolution in the proposed transformer network, the positional encoding is added to the input embeddings to inject the token order information. The positional encoding outputs have the same dimension d_{model} as the embedding outputs so they can sum up.

The position encodings are calculated using the sin and cosine functions as in the following:

$$PE_{(pos,2i)} = sin\left({pos}/{10000^{2i/d_{model}}} \right) \quad (1)$$

$$PE_{(pos,2i+1)} = cos\left({pos}/{10000^{2i/d_{model}}} \right) \quad (2)$$

where pos is the position and i is the dimension? That is, each dimension of the positional encoding corresponds to a sinusoid to get help of its cyclic nature.

The resulting sum vector of each embedding vector and its position encoding vector is fed to the multi-attention head mechanism.

The Transformer Layer

The Transformer consists of a stack of identical encoder layers (see figure 3). Each encoder is composed of two major components: 1) Multi-Head Attention mechanism and 2) Position-wise Feed-Forward Network. What follows is a detailed explanation of each component.

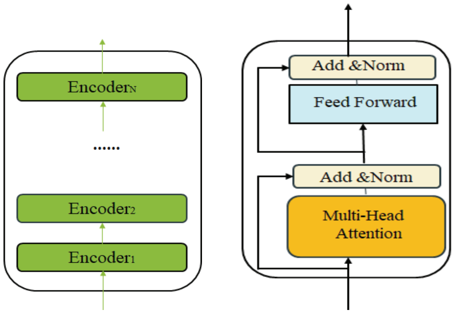

Figure 3. Transformer architecture that consists of several identical encoders (left). Encoder structure (right).

Multi-Head Self-Attention Mechanism

The multi-head attention mechanism, as depicted in the following figure, consists of several attention layers running in parallel. This mechanism has been introduced by Google and uses multiple iterations of computation to capture relevant information. In addition, this component improves the performance of the attention layer in two ways. First, it expands the model's ability to focus on different positions. Second, it gives the attention layer multiple "representation subspaces". The major advantage of Self-Attention is that it ignores the distance between words, computing directly dependency relationships. Thus, making it capable of learning the internal structure of a sentence.

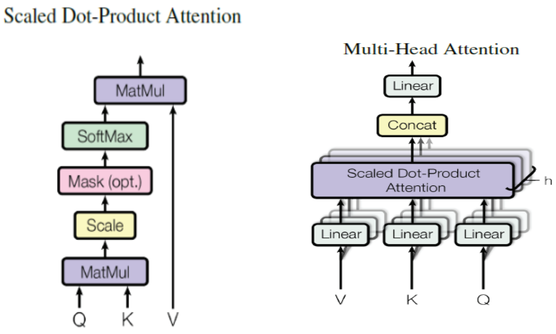

Figure 4. Scaled dot product attention (left). Multi-Head Attention Structure (right).

An attention function consists of mapping a query and a set of key-value pairs to an output, where the query, keys, values and output are all vectors. As mentioned by Vaswani et al (2017), the self-attention calculation can be done using matrices instead of vectors. In this case, we can we compute the multi-head attention as in the following:

$$MultiHead(Q, K, V) = Concat(head_1, head_2, \dots, head_h)W^0 \quad (3)$$

where $$head_i = Attention\ (QW_i^Q, KW_i^K, VW_i^V) \quad (4)$$

$$Attention\ (Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (5)$$

where the queries, keys and values are packed into matrices $Q, K\ and\ V$.

Since the multi-head attention component (see figure 4) is based on multiple attention heads mechanism that runs through the scaled dot-product attention multiple times in parallel. We end up with multiple outputs from each attention head. Since the Feed Forward Network accepts one input, we concatenate the attention heads $(z_1, \dots, z_n)$ associated with each input and then multiply this concatenation with a learned weight matrix W_0. This produces the final outputs Z and $Z'$of the multi-head attention component for the student answer input X and the reference answer input X', respectively.

Position-wise Feed-Forward Networks

The resulting vector of the Multi-Head attention module Z is passed through a fully connected forward network that computes linear transformations of the input. In this work, we consider two 1-dimension convolutions with kernel size d_{inner_head}, a dropout to avoid overfitting and a normalization layer (see Table 2). The dimensionality of input and output is d_{model} .

Layer	Kernel size	Stride
Convolution	$\boldsymbol{d_{inner\_head}}$	1
ReLU	-	-
Convolution	$\boldsymbol{d_{inner\_head}}$	1
ReLU	-	-
Dropout	-	-
Normalization Layer	-	-

Table 2. Feed Forward Network Architecture

Afterwards, we concatenate the outputs of the Feed-Forward Network $[v_1, v_2]$ and pass it to the final SoftMax layer to compute classification probabilities p_1, p_2 for the two classes, correct answer versus incorrect answer.

Experiments

Our experiments were conducted in the context of student freely generated answers in response to hints (in the form of questions) in conversational intelligent tutoring systems. To this end, we have used a previously annotated dataset as described next.

The DT-Grade Dataset

The DT-Grade dataset (Banjade et al. 2016) was created by extracting student responses from logged tutorials interactions between 36 junior level college students and a state of the art conversational ITS. During the interactions, each student solved 9 conceptual physics problems and was asked to provide solutions to each problem in the form of an answer and a full justification based on Physics principles. Their answer was evaluated and if the answer was incorrect or incomplete, e.g., a full justification was not provided, a dialogue followed in which the ITS helped the students to discover the solution through personalized scaffolding in the form of hints that varied in their degree of information/help. Each annotation example in the DT-Grade dataset consists of the following attributes: (1) problem description (describes the scenario or context), (2) tutor question, (3) student answer (without correcting spelling and grammatical errors) and (4) reference answer(s). In addition, the data includes the correctness class of each student answer as judged by a human expert. Each student response was categorized into one of the following four classes: **(1) Correct:** Answer is correct**. (2) Correct-but- incomplete:** The response provided by the student is correct, but something is missing, **(3) Incorrect:** Student answer is incorrect and **(4) Contradictory:** The student answer is contradicting the reference answer**.**

In this work, we consider only two classes: correct and incorrect. The correct answers are those labeled as “correct” in the DT-Grade dataset. All the other instances are considered “incorrect”. As a result, we obtained the following class distribution shown in Table 3 below.

Dataset	Correct (%)	Incorrect (%)
Training	41	59
Testing	41.58	58.41

Table 3. The distribution of classes in training (800 instances) and testing data (100 instances)

Experimental Settings

To evaluate the importance of the components of our attention-based transformer, we have conducted several experiments by varying the transformer architecture and using different embedding approaches.

A first set of experiments have been conducted using word2vec embeddings with 300 dimensions and different settings of the attention-based transformer. Based on the experiments conducted by Vaswani et al. (2017), we have tried various values of the number of attention head (8, 10,15, 32). This has been done to test the impact of increasing and/or decreasing the number of attention heads (*n_head*) on the performance of our model. We have, also, varied the depth of the transformer by experimenting with different number of encoder layers. Other parameters have been modified as well such as the attention key dimension (d_k), the attention value dimension (d_v), and the number of the kernel size (d_{inner_head}) of the convolution layers in the Feed-Forward Network.

Another set of experiments has been conducted using Glove pre-trained embeddings with 300 dimensions. Following the same setup of the first set of experiments, we have used the same values of the major parameters: number of heads of attention, the attention key dimension, the attention value dimension, and the number of layers.

As stated in several research works, ELMo embedding boosted the performance of several deep learning models applied to various NLP tasks. ELMo word vectors are computed on top of a two-layer bidirectional language model (biLM). These biLM layers efficiently encode different types of syntactic and semantic information about words in-context. Using all layers improves overall NLP tasks' performance. For this reason, we have conducted another set of experiments using ELMo embeddings with 1,024 dimensions. We followed the same setup as for the previous sets of experiments and used the same values of the transformer ‘s parameters. The only difference is the value of d_{model} which is set to 1,024 to be consistent with the ELMo embeddings dimension so we can sum up via the position encoding.

Hyperparameters

In all experiments, the model was trained with a categorical cross entropy loss function. For optimization, we used the Adam optimizer with a learning rate of 0.0001, $beta_1$ =0.9 and $beta_2 = 0.99$. The gradients are clipped to 0.5 to prevent exploding gradients. To avoid overfitting, we applied a $dropout = 0.9$ to the sums of the embeddings and the positional encodings of each layer of the transformer. In all experiments, we trained our model for 1,000 epochs to obtain the results. An increasing number of epochs, particularly when using the ELMo embedding, showed an increase in overall accuracy.

Experimental Results

Table 4 shows the accuracy of different architectures of our model using the word2vec embedding. The highest accuracy of 59% was reached when using (15,8,16) heads of attention and (2,6,6) layers, respectively. This result outperforms Bi-GRU-Capsnet with word2vec embeddings (Ait Khayi and Rus 2019) for the same dataset.

d_{model}	d_{inner_head}	n_head	d_k	d_v	$layers$	$accuracy$
300	**512**	**15**	**64**	**64**	**2**	**59**
300	512	10	64	64	2	57
300	2048	8	64	64	1	58
300	2048	32	128	128	8	58
300	**2048**	**8**	**64**	**64**	**6**	**59**
300	512	10	64	64	1	58
300	**4096**	**16**	**128**	**128**	**6**	**59**

Table 4. Results of variations of the transformer architecture using word2vec embeddings.

Table 5 shows results for different architectures of our model using Glove embeddings. The highest accuracy of 60% was reached when using 16 heads of attention and 6 layers of the encoder. It seems that increasing the number of heads of attention above 16 has led to a decrease in accuracy. This led us to our first observation: there are specific heads of attention that play an important role in the transformer and a specific number of these heads of attention is sufficient to achieve good results. Thus, adding more heads of attention can be considered redundant for the transformer ‘s architecture. This performance is better than the performance obtained with the word2vec embeddings and outperforms the approach based on Bi-GRU-Capsnet with Glove embeddings.

d_{model}	d_{inner_head}	n_head	d_k	d_v	$layers$	$accuracy$
300	512	15	64	64	2	59
300	512	10	64	64	2	57
300	2048	8	64	64	1	56
300	2048	32	128	128	8	56
300	2048	8	64	64	6	56
300	512	10	64	64	1	58
300	**4096**	**16**	**128**	**128**	**6**	**60**

Table 5. Results using Glove embeddings.

Table 6 shows results for different architectures of our model using ELMo embeddings. We can observe an improvement in the overall accuracy in comparison with the results obtained with the Glove and Word2vec embeddings. The highest accuracy of 71.5% was achieved when using 10 heads of attention and 2 layers only. This is a very competitive result that compares with the Bi-GRU-Capsnet approach in combination with ELMo embeddings (72.5%).

d_{model}	d_{inner_head}	n_head	d_k	d_v	$layers$	$accuracy$
1024	512	15	64	64	2	61
1024	**512**	**10**	**64**	**64**	**2**	**71.5**
1024	2048	8	64	64	1	64
1024	2048	32	128	128	8	61
1024	2048	8	64	64	6	61
1024	512	10	64	64	1	67

Table 6. Results using ELMo embeddings.

The results provided in Table 7 show that the attention-based transformer outperformed the Bi-GRU Capsnet based approach when using Glove embeddings: 65% versus 56.25 % accuracy, respectively. It has outperformed also the Bi-GRU Capsnet when using the Word2vec embeddings: 61 % versus 56.25%. The highest accuracy of 71.5% was achieved with ELMo embeddings. The results show also that our proposed model displays a superior performance over the baseline models: Bi-GRU and LSTM. An interesting finding in the conducted experiments is that the proposed attention model handles the assessment of short answers with a small number of words less than 6 better than the recurrent networks: Bi-GRU and LSTM. This can be explained by the fact that the self-attention mechanism in our proposed model allows the selection of the most relevant words in the students answer and reference answer while also accounting for the larger context provided by the problem description and the previous tutor question. Then, the similarity score is computed based on those relevant words. For example, giving the following reference answer “The ball is slowing down at a constant rate “, the attention mechanism allows to focus on the most relevant part of this input: “slowing down”. This selected part has a similar semantic representation with the following student answer: “it is decreasing”. Thus, the transformer is capable to assess this answer correctly. Similar observations were made for other short student answers with fewer words

Model	Accuracy
Transformer (ELMo)	71.5
Bi-GRU-Capsnet (ELMo)	**72.5**
Transformer (Glove)	**60**
Bi-GRU-Capsnet (Glove)	56.25
Transformer (word2vec)	**59**
Bi-GRU-Capsnet (Word2vec)	56.25
Bi-GRU	56.25
LSTM	60

Table 7. Comparison with other deep learning models

Conclusion

In this paper, we proposed an attention-based transformer to assess the correctness of student answers freely generated by student in dialogue-based ITSs. To the best of our knowledge, this is the first time an attention-based approach has been applied to the task of assessing the correctness of student responses. The proposed approach was chosen due to promising results that transformers achieved in various NLP tasks, especially in semantic similarity and text classification. Furthermore, adding attention to deep learning models has led to a significant gain in their performance. Experimental results on the DT-Grade dataset show high competitiveness of the proposed model, rivalling previously proposed state-of-the-art methods. The highest accuracy obtained was 71.5% using ELMo embeddings which is very close to the best results achieved by Bi-GRU-Capsnet (72.5) on the DT-Grade dataset. The main advantage of our proposed model over the Bi-GRU-Capsnet is reducing the training time and giving more space for parallelization. As a future work, we will keep applying novel deep learning to improve our current results.

Acknowledgments. This research was partially sponsored by the UofM, the Institute for Education Sciences under award R305A100875 to Dr. Vasile Rus and IIS.

References

Ait Khayi, N., & Rus, V. 2019. BI-GRU Capsnet for Student Answers Assessment. DL4Ed workshop at the 26th ACM SIGKDD.

Banjade, R., Maharjan, N., Niraula, N. B., Gautam, D., Samei, B., & Rus, V. 2016. Evaluation Dataset (DT-Grade) and Word Weighting Approach Towards Constructed Short Answers Assessment in Tutorial Dialogue Context. In Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications (pp. 182-187). doi. 10.18653/v1/W16-0520

Bojanowski, P.; Grave, E.; Joulin, A.; and Mikolov, T. 2017. Enriching Word Vectors with Subword Information.Transactions of the Association for Computational Linguistics 5:135–146. doi:10.1162/tacl_a_00051

Courtney Corley and Rada Mihalcea. 2005. Measuring the Semantic Similarity of Texts. In Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment. doi:10.3115/1631862.1631865

Dikli, S.2006. An Overview of Automated Scoring of Essays. *The Journal of Technology, Learning and Assessment* 5(1). doi.org/10.4135/9781483346397.n35

Maharjan, N., Gautam, D., & Rus, V. 2018. Assessing Free Student Answers in Tutorial Dialogues Using LSTM Models. In the proceedings of AIED. doi:10.1007/978-3-319-93846-2_35

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., & Dean, J. 2013. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of NIPS.

Nenkova, A., & McKeown, K. 2011. Automatic summarization. Foundations and Trends in Information Retrieval, 5(2–3), 103-233.

Ngoc Phuoc An Vo, Simone Magnolini, and Octavian Popescu. 2015. FBK-HLT: An Application of Semantic Textual Similarity for Answer Selection in Community Question Answering. In Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval. volume 15, pages 231–235. doi:10.18653/v1/S15-2041

Pennington, J., Socher, R., & Manning, C.D. 2014. Glove: Global Vectors for Word Representation. In Proceedings of EMNLP. doi:10.3115/v1/D14-1162

Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C.R., Lee, K., & Zettlemoyer, L.S. 2018. Deep Contextualized Word Representations. arXiv preprint arXiv:1802.05365

Rus, V., D’Mello, S. K., Hu, X., & Graesser, A. C. 2013. Recent advances in intelligent tutoring systems with conversational dialogue, *AI Magazine*, 34(3), 42-54.

Rus, V., Niraula, N.B., and Banjade, R. 2015. DeepTutor: An Effective, Online Intelligent Tutoring System that Promotes Deep Learning. In Proceedings of AAAI.

Tang, X., Chen, S., Do, L., Min, Z., Ji, F., Yu, H., Zhang, J., & Chen, H. 2018. Improving Multilingual Semantic Textual Similarity with Shared Sentence Encoder for Low-resource Languages. arXiv preprint arXiv:1810.08740.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., & Polosukhin, I. 2017. Attention Is All You Need. In Proceedings of NIPS.

Wang, Q., Ruan, T., Zhou, Y., Xu, C., Gao, D., & He, P. 2018. An Attention-based BI-GRU-CapsNet Model for Hypernymy Detection between Compound Entities. In Proceedings of 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 1031-1035. doi:10.1109/BIBM.2018.8621408

Wang, Y., Huang, M., Zhu, X., & Zhao, L. 2016. Attention-based LSTM for Aspect-level Sentiment Classification. In Proceedings of EMNLP. doi:10.18653/v1/D16-1058

Yang, B., Tu, Z., Wong, D.F., Meng, F., Chao, L.S., & Zhang, T. 2018. Modeling Localness for Self-Attention Networks. In Proceedings of EMNLP. doi:10.18653/v1/d18-1475