UPB at SemEval-2021 Task 8: Extracting Semantic Information on Measurements as Multi-Turn Question Answering

Andrei-Marius Avram[1,2], George-Eduard Zaharia[1], Dumitru-Clementin Cercel[1], Mihai Dascalu[1]
University Politehnica of Bucharest, Faculty of Automatic Control and Computers[1]
Research Institute for Artificial Intelligence, Romanian Academy[2]
{andrei_marius.avram, george.zaharia0806}@stud.acs.upb.ro
{dumitru.cercel, mihai.dascalu}@upb.ro

Abstract

Extracting semantic information on measurements and counts is an important topic in terms of analyzing scientific discourses. The 8th task of SemEval-2021: Counts and Measurements (MeasEval) aimed to boost research in this direction by providing a new dataset on which participants train their models to extract meaningful information on measurements from scientific texts. The competition is composed of five subtasks that build on top of each other: (1) quantity span identification, (2) unit extraction from the identified quantities and their value modifier classification, (3) span identification for measured entities and measured properties, (4) qualifier span identification, and (5) relation extraction between the identified quantities, measured entities, measured properties, and qualifiers. We approached these challenges by first identifying the quantities, extracting their units of measurement, classifying them with corresponding modifiers, and afterwards using them to jointly solve the last three subtasks in a multi-turn question answering manner. Our best performing model obtained an overlapping F1-score of 36.91% on the test set.

1 Introduction

Our world revolves around quantities and units of measurement present in all texts, ranging from scientific texts to recipes. Nevertheless, the process of automatically extracting measurements is not trivial, considering that, in most situations, the quantitative structures are ambiguous and are not present in the same area within the text. Therefore, parsing the semantic relations becomes a ubiquitous task, since proper quantity identification leads to transformations towards an easy to follow quantitative summary. Advantages of the previously mentioned process can be found in medical prescriptions (Adamo et al., 2015). As such, a system that can robustly and confidently identify medication quantities, measurement units, as well as the medication itself has the potential to become a breakthrough for computer-based medicine and consultations. Another use case resides in ERP systems where proper parsing of resource descriptions facilitates the identification of similar or duplicate items.

The MeasEval - Counts and Measurements competition (Harper et al., 2021) organized by the 15th International Workshop on Semantic Evaluation (SemEval-2021) creates a new challenge in the area of Natural Language Processing, proposing five subtasks related to span identification, classification, as well as relation extraction, that aim to improve the state of the art for the current field of measurement information extraction. We created a cascaded system to solve the stated problem that is composed of: (1) a subsystem that identifies quantities in the input text; (2) a subsystem that classifies their value modifiers; (3) a subsystem that extracts their measurement unit; and (4) a subsystem that then finds the appropriate measured entities, measured properties, and qualifiers by asking questions related to entity-relations. Three pretrained Transformer-based (Vaswani et al., 2017) language models are experimented for subsystems (1) and (4) of the cascaded system by fine-tuning them on the specific task: Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019), Robustly Optimized BERT Pretraining Approach (RoBERTa) (Liu et al., 2019), and Science BERT (SciBERT) (Beltagy et al., 2019). A character-level bidirectional Long Short-Term Memory (BiLSTM) (Hochreiter and Schmidhuber, 1997) architecture was considered for subsystems (2) and (3).

The rest of the paper is structured as follows. The next section presents a series of solutions associated with relation extraction, span identification,

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 534–540
Bangkok, Thailand (online), August 5–6, 2021.

and measurement unit identification. The third section outlines our approaches related to the subtasks proposed by the competition. The fourth section presents a performance evaluation of our systems together with an error analysis, while the final section concludes our work and outlines potential future improvements.

2 Related Work

Span Identification. Papay et al. (2020) studied the performance of various models designated for different span identification tasks. Out of them, we mention Conditional Random Fields (CRFs) (Lafferty et al., 2001), LSTM cells with CRF, BERT+CRF, LSTM+BERT+CRF, or handcrafted features, usable with any of the previously mentioned models. At the same time, a language model was specifically developed for the span identification tasks, entitled SpanBERT (Joshi et al., 2020), by masking an entire sequence instead of masking a single word in its pretraining process. The authors argued that SpanBERT obtained substantial gains on span selection tasks, such as question answering and coreference resolution.

Measurement Unit Identification. Berrahou et al. (2013) proposed a two-step system for search space size reduction, followed by unit extraction from the previously obtained textual fragments. Also, Hundman and Mattmann (2017) presented a hybrid system composed of a CRF that identifies quantities values and their units, followed by a rule-based model to detect their corresponding entities.

Relation Extraction. Zhang and Wang (2015) adopted a model based on Recurrent Neural Networks (RNN) (Cho et al., 2014) composed of three main elements: an embedding layer, a bidirectional recurrent layer, followed by a max pooling layer that produces the feature vector used for relation classification. RNN-based models were also applied by Zhang et al. (2015) who adopted BiLSTMs, or by Xiao and Liu (2016) who proposed an architecture based on hierarchical RNNs alongside an attention mechanism. Furthermore, several convolutional neural network-based models with various approaches were proposed, for example: multi-level attention (Wang et al., 2016), attention-based context vectors (Shen and Huang, 2016), or multi-level features (word, lexical, sentence) (Zeng et al., 2014). BiLSTMs are also present in the work of Lee et al. (2019) who implemented a mechanism based on entity-aware attention using latent entity typing. Jin et al. (2020) approached the relation extraction task by employing a Graph Neural Network system that modeled each relation as a node and learned the dependencies between the nodes.

3 Method

Our approach on MeasEval consisted of a cascade system composed of individual subsystems for each of the problems in the first two subtasks, and then jointly solving the last three subtasks with a single subsystem.

3.1 Quantity Identification

The subtask of identifying quantities in text was formalized as a sequence labeling problem with Inside–Outside–Beginning (IOB) tags (Ramshaw and Marcus, 1999) that were predicted by a pretrained language model with a CRF on top of predicted logits, as proposed by Avram et al. (2020). The architecture is depicted in Figure 1.

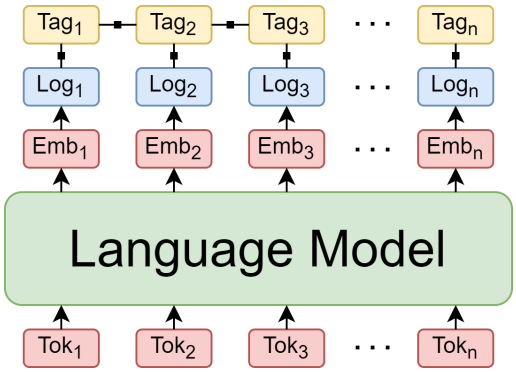

Figure 1: Quantity identification subsystem architecture.

More formally, we project each output embedding e_i produced by the pretrained language model into probability logits l_i by using a feed-forward network with a ReLU activation as $l_i = ReLU(W_l^T e_i + b_l)$, where W_l is the corresponding weight matrix and b_l is the corresponding bias. Then, we model the output conditional probabilities for each tag y_i by using the CRF learning algorithm, as depicted in Eq. 1:

$$p(y|l) = \frac{1}{Z} exp \left\{ \sum_{i=1}^{n} W_{y_{i-1},y_i}^T l_i + b_{y_{i-1},y_i} \right\} \quad (1)$$

where W_{y_{i-1},y_i} and b_{y_{i-1},y_i} are the weight matrix and the bias of the CRF, and Z is a normalization constant such that the probabilities sum up to one.

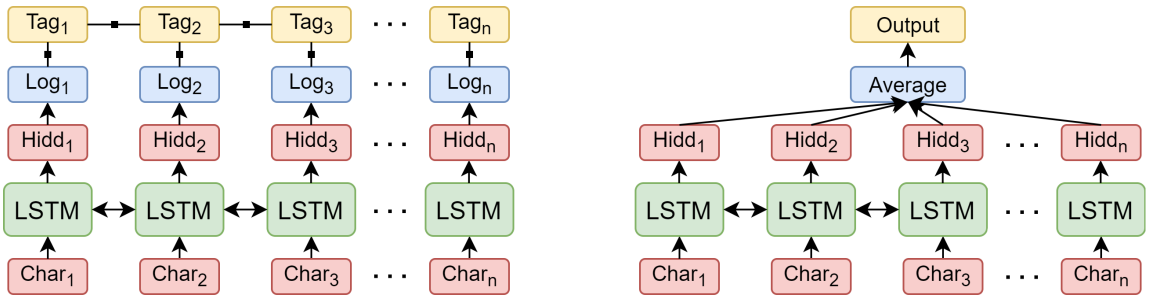

Figure 2: Architectures used in unit extraction (left) and value modifiers classification (right) subsystems.

The entire subsystem is trained to maximizing the log-likelihood of the data, while the Viterbi algorithm (Forney, 1973) is used during inference to find the most likely sequence of tags.

3.2 Unit Extraction and Value Modifier Classification

For the second subtask, a character-level BiLSTM extracts the units from quantities and classifies their corresponding value modifiers. We approached the unit extraction in a similar way as the quantity identification, by treating the problem as a sequence tagging; however, the pretrained language model was replaced with BiLSTM cells. Moreover, instead of predicting a label for each character (token), we instead averaged the BiLSTM hidden states and projected their average in an eleven-dimensional vector (i.e., number of possible value modifiers) for the classification. Then, a sigmoid activation function was applied to obtain a vector that contains the probability of the quantity to belong to a class at each index. The architectures used for unit extraction and value modifiers classification are depicted in Figure 2.

3.3 Joint Entity Identification and Relations Extraction

Subtask Grouping. The last three subtasks were grouped into a single subtask where a pretrained language model was fine-tuned to jointly identify three elements: the span of the measured entities, the measured properties, and their corresponding qualifiers. The model extracts the relations between the three elements and the previously extracted quantities using a multi-turn question answering (QA) architecture, as proposed by Li et al. (2019). The pretrained language models used for this task were identical to the ones from the quantity identification subtask.

Question Templates. The input to the subsystem is created by appending a question before the text that denotes a possible relation between a given and a target entity. There are a total of six question templates that can be filled with the corresponding entities that cover all the possible relations, as depicted in Table 1. Then, the questions are asked in a specific order to correctly identify the relations and the span of the entities. First, starting with a given quantity, the model is asked to identify its measured properties. If a measured property is found, the model marks its span and links it to the quantity with the `HasQuantity` relation (question 1). Second, the model is asked to identify the measured entity with that corresponding measured property, linking the two with the `HasProperty` relation (question 2). Third, if no measured property is found for a given quantity, the model is asked to directly identify the measured entity, marking the relation between the measured entity and the quantity directly with `HasQuantity` (question 3). Finally, once all quantities, measured entities, and properties are identified, the model is asked to identify corresponding qualifiers and marks the relations accordingly (questions 4-6 in table).

Model Output. The architecture proposed in (Devlin et al., 2019) for SQuAD 2.0 (Rajpurkar et al., 2018) is employed to create the output of the subtasks; as such, two vectors are used for fine-tuning: a starting vector S and an ending vector E. The probability of token i to be the start of a span is computed as a dot-product between the embedding T_i and the start vector S, followed by a softmax applied over all the tokens of the input: $P = softmax(T_i \cdot S)$. An analogous formula computes the end probabilities of a span. Then, we take the indices i and j are taken to compute the most probable span for an entity, where $i \leq j$ maximizes the sum of log-likelihoods $T_i \cdot S + T_j \cdot E$. We compare for each query the previously defined maximum sum with the sum of the start and end log-likelihoods of the `[CLS]` token s_{null} because

#	Relation Type	Question
1	HasQuantity	What is the *measured property* of the *quantity* ____?
2	HasProperty	What is the *measured entity* that has the *measured property* ____ of the *quantity* ____?
3	HasQuantity	What is the *measured entity* that has the *quantity* ____?
4	Qualifies	What is the *qualifier* corresponding to the *quantity* ____?
5	Qualifies	What is the *qualifier* corresponding to the *measured entity* ____?
6	Qualifies	What is the *qualifier* corresponding to the *measured property* ____?

Table 1: Question templates for each relation type.

there can be questions without an answer[1]. If this sum is higher, then there is no such type of relationship for that entity. A threshold added to the s_{null} is considered in order to provide a higher granularity between questions with or without answers, which was tuned on the development set to maximize F1-score.

Figure 3 introduces our architecture for entity recognition and relation extraction. The question tokens marked with `Qst` and the paragraph tokens marked with `Tok` are fed as input, while the start `S` and the end `E` logits are present at output.

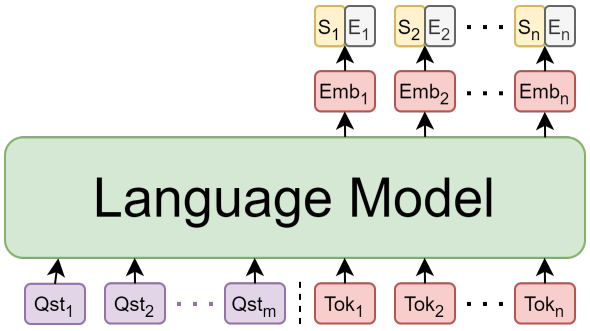

Figure 3: Joint entity recognition and relation extraction architecture as multi-turn question answering.

4 Performance Evaluation

4.1 Experimental Setup

Dataset Analysis and Processing. The provided corpus for the competition was quite scarce, counting 298 samples in both train and trial datasets. The corpus contained texts from the scientific domain, counting a total of 7,979 unique words with an average sentence length of approximately 160 words. For training our models, we merged the train and trial subsets and randomly split them into 90% training and 10% development.

Pretrained Language Models. An Adam optimizer (Kingma and Ba, 2014) with a learning rate of 2e-5 was used for training the subsystem of the first and last three subtasks. We experimented with the large versions of BERT and RoBERTa, and with the base version of SciBERT because there are currently no implementations available online of its large variant. Each subsystem was fine-tuned for 10 epochs; the subsystems that employed large language model variants were trained with a batch size of 2 due to the computational constraints, whereas the subsystems that employed base language model variants used a batch size of 8.

BiLSTM Networks. An Adam optimizer was also employed for training the BiLSTM networks of the second subtask, but with a learning rate of 1e-4. The models were trained for 25 epochs using a batch size of 16. We stacked the LSTM cells two times and used a hidden size of 64, with an embedding of 32 dimensions for the characters.

4.2 Results

The results of each subsystem on the development set are introduced in Table 3. SciBERT obtained the highest F1-score on both quantity identification and joint entity and relation extraction, although it is smaller when compared with the other two models. On the second subtasks, the model achieved a reasonable performance, 95.75% F1-score on unit extraction and 88.94% F1-score on value modifier classification.

The results of the cascaded system are presented in Table 2 that introduces the global precision, recall, and F1-scores averaged across all subtasks, as well as the exact match (EM) and overlap F1 scores[2] between the gold annotations and our predictions. As opposed to the performance of each subsystem on the development set where SciBERT was the best performing model, RoBERTa obtained the highest scores as a whole system, with an overlap F1-score of 39.05% on the development set and 36.91% on the test set, outperforming SciBERT

[1]Only measured properties and qualifiers related questions are allowed to not have an answer. Measured entity–related questions must always have an answer.

[2]The overlap F1-score was the metric by which the competition systems were ranked.

	Avg. Precision		Avg. Recall		Avg. F1		EM		Overlap F1	
System	**Dev**	**Test**	**Dev**	**Test**	**Dev**	**Test**	**Dev**	**Test**	**Dev**	**Test**
BERT-related	**61.10**	58.69	52.28	48.05	56.63	52.84	31.61	25.58	36.72	32.69
RoBERTa-related	60.04	**61.01**	**56.05**	**52.66**	**58.14**	**56.53**	**34.98**	**30.89**	**39.05**	**36.91**
SciBERT-related	56.73	54.35	54.61	46.12	55.65	49.90	30.82	23.71	35.89	30.30

Table 2: Results averaged across all five subtasks on the development and test sets.

Subsystem	**Precision**	**Recall**	**F1**
Quantity identification			
RoBERTa-CRF	90.77	92.85	91.26
BERT-CRF	**91.77**	93.72	92.38
SciBERT-CRF	91.60	**95.02**	**93.00**
Unit extraction and value modifier classification			
Unit Extraction	96.44	95.27	95.75
Value Modifiers	91.82	86.65	88.94
Joint relation extraction and entity identification			
RoBERTa-QA	71.04	71.26	71.14
BERT-QA	72.18	**71.09**	71.63
SciBERT-QA	**73.81**	70.71	**72.22**

Table 3: Performance analysis on the development set.

with over 6% and over 3%, respectively. More surprisingly, SciBERT also obtained a lower score than BERT on both sets, having an overlap F1-score lowered by 2% and 1%. We believe that these differences between the scores of RoBERTa and SciBERT were caused by the way the two models were evaluated as stand-alone subsystems or as a whole system.

4.3 Error Analysis

Quantity Sensitivity. The main drawback in our approach was that all other subtasks were highly dependent on the quality of the extracted quantities for the first subtask. To exemplify this, let us consider the case where a modifier like *"approximate"* is missed before a quantity; afterwards, it would be impossible to correctly classify its modifiers. Another use case is when the subsystem misses the measuring unit, with the same effect on the unit extractor. Moreover, we noticed that the joint entity and relation extraction was especially sensible to partially identified quantities, producing mostly bad outputs in these cases.

Measured Unit Inference. Another limitation of our approach emerges when the unit extractor does not identify all units in a sequence tagging style. This happened in cases when the unit was split across several places in the quantity, or when it had to be predicted from the context. For instance, the correct unit would be *"m^2"* when encountering a *"300 m x 400 m"* quantity; however, our model found only *"m"* as unit.

Long Documents. Finally, several documents had a longer sequence length than 512 tokens[3] when tokenized, which surpasses the maximum admitted length by the pretrained language models; the workaround was to simply remove the tokens after position 512. However, this solution has the obvious effect of missing identifiable entities that appear after this position.

5 Conclusions and Future Work

This paper introduces our approach that solves all the five subtasks of the 8th task of SemEval-2021 competition in a cascaded manner. First, quantities are identified as a sequence tagging task by using a pretrained language model with a CRF layer. Then, the measurement units are extracted and the modifiers are classified using BiLSTMs at character level on the identified quantities. Finally, the measured entities, measured properties, and qualifiers are jointly identified, together with their relations, by using a multi-turn question answering approach with hand-crafted questions specific to each relation type. Our best model obtained an F1-score of 36.91% on the test set. We further emphasized several limitations of our approach and showed that the overall performance was highly sensitive to the quality of the identified quantities.

A possible direction for future work is to test the system using language models that can process longer sequences, such as Longformer (Beltagy et al., 2020) or BigBird (Zaheer et al., 2020), in order to reduce the effect of missing entities simply due to the sequence length. We also consider creating an ensemble model using several pretrained language models to boost the overall performance, as reported by Ionescu et al. (2020).

[3] Approximately 4% of the documents had more than 512 tokens for each pretrained language model.

References

Francesco Adamo, Filippo Attivissimo, Attilio Di Nisio, and Maurizio Spadavecchia. 2015. An automatic document processing system for medical data extraction. *Measurement*, 61:88–99.

Andrei-Marius Avram, Dumitru-Clementin Cercel, and Costin Chiru. 2020. UPB at SemEval-2020 task 6: Pretrained language models for definition extraction. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 737–745.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3606–3611.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Soumia Lilia Berrahou, Patrice Buche, Juliette Dibie-Barthelemy, and Mathieu Roche. 2013. How to extract unit of measure in scientific documents? In *KDIR: Knowledge Discovery and Information Retrieval*, pages 454–459. Springer.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

G David Forney. 1973. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278.

Corey Harper, Jessica Cox, Curt Kohler, Antony Scerri, Ron Daniel Jr., and Paul Groth. 2021. SemEval 2021 task 8: MeasEval – extracting counts and measurements and their related contexts. In *Proceedings of the Fifteenth Workshop on Semantic Evaluation (SemEval-2021)*, Bangkok, Thailand (online). Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Kyle Hundman and Chris A Mattmann. 2017. Measurement context extraction from text: Discovering opportunities and gaps in earth science. *arXiv preprint arXiv:1710.04312*.

Marius Ionescu, Andrei-Marius Avram, George-Andrei Dima, Dumitru-Clementin Cercel, and Mihai Dascalu. 2020. UPB at FinCausal-2020, tasks 1 & 2: Causality analysis in financial documents using pretrained language models. In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, pages 55–59.

Zhijing Jin, Yongyi Yang, Xipeng Qiu, and Zheng Zhang. 2020. Relation of the relations: A new paradigm of the relation extraction problem. *arXiv preprint arXiv:2006.03719*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

Joohong Lee, Sangwoo Seo, and Yong Suk Choi. 2019. Semantic relation classification via bidirectional lstm networks with entity-aware attention using latent entity typing. *Symmetry*, 11(6):785.

Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019. Entity-relation extraction as multi-turn question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1340–1350.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Sean Papay, Roman Klinger, and Sebastian Padó. 2020. Dissecting span identification tasks with performance prediction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4881–4895.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789.

Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer.

Yatian Shen and Xuan-Jing Huang. 2016. Attention-based convolutional neural network for semantic relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2526–2536.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010.

Linlin Wang, Zhu Cao, Gerard De Melo, and Zhiyuan Liu. 2016. Relation classification via multi-level attention cnns. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1298–1307.

Minguang Xiao and Cong Liu. 2016. Semantic relation classification via hierarchical recurrent neural network with attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1254–1263.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *arXiv preprint arXiv:2007.14062*.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers*, pages 2335–2344.

Dongxu Zhang and Dong Wang. 2015. Relation classification via recurrent neural network. *arXiv preprint arXiv:1508.01006*.

Shu Zhang, Dequan Zheng, Xinchen Hu, and Ming Yang. 2015. Bidirectional long short-term memory networks for relation classification. In *Proceedings of the 29th Pacific Asia conference on language, information and computation*, pages 73–78.