Improving a Multi-Head Selection Model for Multiple-Relations Extraction

Peiyong Liu 260765100
Nikhil Krishna 260862308
Niles Jiang 260789242

Abstract

Multiple-relations extraction (MRE) is the process of taking an input paragraph and identifying relationships between two or more entity mentions in the text. While most state-of-the-art solutions for MRE require multiple-pass encoding on the input corpus, the model proposed by (Wang et al., 2019) can complete MRE with one-pass encoding. In this paper, we present our experiment testing the proposed model on both an English dataset and a Chinese dataset. Additionally, we further improve the performance of the existing model by using ALBERT$_{xlarge}$ embeddings rather than BERT$_{base}$ embeddings. We see that although this keeps the training time of the overall model the same, the overall accuracies are increased by a few percentage points.

1 Introduction

Multiple-relations extraction (MRE) is designed to identify relations among multiple pairs of entity mentions from an input paragraph. Most approaches to MRE require multiple passes to encode pairs of entities because they consider each pair as an independent instance. Clearly, this method is more computationally expensive than simply doing one-pass.

1.1 Relation Extraction

Relation extraction (RE) is the task of extracting semantic relationships from text. Extracted relations usually occur between two or more entities of a certain type (e.g. Person, Organisation, Location) and fall into a number of semantic categories (e.g. married to, employed by, lives in). RE can be denoted using triples, {subject_type, relation, object_type}. For instance, given the sentence "Paris is in France", the two entities are "Paris", the subject, and "France", the object. Both of their types are Locations. The sentence states an "is in" relationship from "Paris" to "France", thus RE can be written as Location, is in, Location. Another example is "Barack Obama was born in Hawaii". In this case, the subject is "Barack Obama", whose type is Person, and the object is "Hawaii", whose type is Location. The relationship is "born in" from "Barack Obama" to "Hawaii". Thus, the RE triple can be written as {Person, born in, Location}.

2 Model

2.1 Original Model

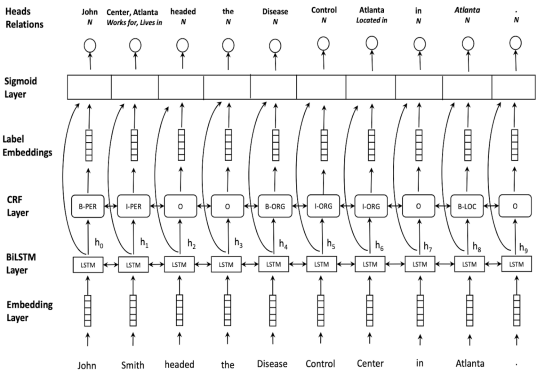

Figure 1: A visual representation of the original model.

This work consists of two models, an original from previous work (Bekoulis et al., 2018) and our modified version. The original model we use is taken from (Bekoulis et al., 2018) and is a multi-head selection model, i.e. any entity can be involved in a number of relations with other entities. The sentence's words are first fed into an embedding layer that creates a vectorized representation. Then this representation is passed through BiLSTM and CRF layers, and this output is finally fed into a sigmoid layer. Figure 1 shows the model along with a sentence and its output for an example. For each word, two outputs are given. The first is an entity recognition label and the second is a set that contains an entity token and the relation between the previous entity label and this one. This is shown in the figure. Sometimes, words in a sentence are

part of no relation. These are labelled with "N" and the predicted head is itself (Bekoulis et al., 2018).

For the embedding layer, we decide word embeddings using the Skip-Gram word2vec pre-trained model from (Mikolov et al., 2013). However, other embeddings such as BERT-based vectorizations were also used in this model to capture certain character-level features such as prefixes. Note that we want to extract multiple-relations in a single pass. To this end, we use BERT$_{base}$ to make the embedding process more efficient and able to compute the relations in this manner. Furthermore, each character is transformed into a vector and fed to a BiLSTM layer, which concatenates the forward and backward state to produce a new, better representation of the character (Bekoulis et al., 2018). Finally, this embedding is combined with the word2vec and BERT embeddings to produce the result.

The purpose of the CRF layer is to perform the named entity recognition task using a Beginning, Inside, Outside (BIO) encoding. For each entity, we have multiple tokens, each for which we should be able to produce a tag. To this end, assign a B to the first token of the entity, an I to an inner token, and an O to a token that doesn't belong to a relation. Figure 1 shows an example of this being done on "John Smith." John is given a B-PER tag, indicating it is the first token of the entity and the entity is a person and Smith is given an I-PER tag, indicating it is an inner token of the person entity. The CRF layer is used to find the named tags for each token as well as the boundaries. The score for each tag is given by Equation 1.

$$s(h_i) = V \cdot f(Uh_i + b) \tag{1}$$

Here, f is some activation function, $V_e \in \mathbb{R}^{p \times l}$, $U_e \in \mathbb{R}^{l \times 2d}$, $b_e \in \mathbb{R}^l$, where d is the hidden size of the LSTM, p is the number of possible tags, and l is the width of the layer. Using this methodology, we determine the sequence of tags that has the highest score for a sentence.

The relation extraction task is formulated as a multi-head selection problem, i.e. each token can be a part of multiple relations. Given a token, a 2-tuple is predicted. The first element is the vector of predicted heads for the token, and the second is the vector of possible relations that the token is predicted to be a part of (Bekoulis et al., 2018). The score for a token w_j and a relation r_k given some token w_i is

$$s(w_j, r_k \mid w_i) = V \cdot f\left(Uw_j + Ww_i + b\right) \tag{2}$$

where f is an activation function, $V_r \in \mathbb{R}^l$, $U_r, W_r \in \mathbb{R}^{l \times (2d+b)}$, $b_r \in \mathbb{R}^l$, d is the hidden size of the LSTM, l is layer width, and b is the dimension of the label embedding (Bekoulis et al., 2018).

Finally, this value can now be used to compute the probability shown in Equation 3

$$\Pr(w_j, r_k \mid w_i) = \sigma(s(w_j, r_k \mid w_i)) \tag{3}$$

Equation 3 represents the probability that w_j is selected as the head token of w_i, and there is an r_k relation predicted between them (Bekoulis et al., 2018). Note that we use the sigmoid function to map the scores to a value between 0 and 1. This probability can then be used to compute the training loss function,

$$\mathcal{L} = -\sum_{i=0}^{n}\sum_{j=0}^{m} \Pr(w_j, r_k \mid w_i) \tag{4}$$

where n is the number of training examples and m is the number of relations w_i is a part of. During application, the w_j and r_k were the actual (not predicted) values of the head and relation, respectively (Bekoulis et al., 2018).

2.2 Updated Model

In this paper, we improve the original model by changing BERT$_{base}$ to ALBERT$_{xlarge}$, which uses 44.4% fewer parameters than the state-of-the-art BERT model, with little loss of accuracy (Lan et al., 2020). This parameter-size reduction provides the opportunity to scale up again. In addition, this property of ALBERT also significantly shortens the training time required. Tables 1 and 2 illustrate the aforementioned improvement of ALBERT. We keep the other architectures of the model to be the unchanged.

Model		Parameters	Layers	Hidden	Embedding	Parameter-sharing
BERT	base	108M	12	768	768	False
	large	334M	24	1024	1024	False
ALBERT	base	12M	12	768	128	True
	large	18M	24	1024	128	True
	xlarge	60M	24	2048	128	True
	xxlarge	235M	12	4096	128	True

Table 1: The configurations of the main BERT and ALBERT models analyzed in this paper (Lan et al., 2020)

Models	Steps	Time	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
BERT-large	400k	34h	93.5/87.4	86.9/84.3	87.8	94.6	77.3	87.2
ALBERT-xxlarge	125k	32h	**94.0/88.1**	**88.3/85.3**	87.8	**95.4**	**82.5**	**88.7**

Table 2: The effect of controlling for training time, BERT-large vs ALBERT-xxlarge configurations (Lan et al., 2020)

3 Related Work

Feature-based joint models (Kate and Mooney, 2010; Yang and Cardie, 2013; Li and Ji, 2014; Miwa and Sasaki, 2014) have been proposed to solve the entity recognition and RE sub-tasks simultaneously. However, these models put a heavy burden on the data preprocessing phase since they rely on the manually designed features and availability of NLP tools (e.g., POS taggers). The joint neural network model we adopted (Bekoulis et al., 2018) overcomes this issue by automatically performing end-to-end relation extraction without any need of manual features or the use of additional NLP components.

Most existing MRE solutions are based on feature or model architecture selection techniques, or domain adaptation approaches. However, these approaches are mainly variations on single relation extraction (SRE) approaches, and therefore require multiple passes of encoding over the paragraph.

The model presented also adopts the One-Pass MRE model proposed by (Wang et al., 2019), which proposed a state-of-the-art solution to complete the MRE task with only one-pass encoding. The one-pass encoding structure also enables it to scales to a larger dataset easily. The model uses Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) as the transformer-based encoder to perform one-pass encoding.

After the release of BERT in the year 2019, there have been many off-springs of BERT coming out, outperforming BERT on various tasks (Sanh et al., 2020; Lan et al., 2020; Sun et al., 2020). A Lite BERT (ALBERT) is a BERT variant designed by Google to improve accuracy training time (Lan et al., 2020). We choose to use ALBERT$_{xlarge}$ to replace BERT$_{base}$ used in the original model as it still has fewer parameters (60M) than BERT$_{base}$ (108M) but will perform the best of all ALBERT variants for which this is the case.

4 Methods

4.1 Datasets

The Chinese dataset comes from the original development team. Originally, the full dataset contained a total of 273,105 datapoints. We remove most of these to reduce training time when training on the full dataset. The total number of datapoints was reduced to 22,944 by removing random points. The data was split so that a 70-30 ratio among the training and validation/test data was present. There are 50 relations available in total. Each schema has three keys: subject_type, predicate, and object_type, where subject_type and object_type are entities, and predicate are relations between the two entities. Both train_set and test_set are also provided. In both datasets, sentences have been POS tagged already. This helped us as the relations can be directly determined by applying the model. A training example is shown below in Figure 2.

{"postag": [{"word": "广东天联集团有限公司", "pos": "nt"}, {"word": "成立", "pos": "v"}, {"word": "于", "pos": "p"}, {"word": "1993年", "pos": "t"}], "text": "广东天联集团有限公司成立于1993年"}

Figure 2: An example of a data point used in training

The English dataset comes from (Batista, 2020). There are 22,944 words total coming from tokenized sentences scraped from Wikipedia articles. A part of the raw, original data is shown below in Figure 3.

4	1919	ADJ	I–NP	O
5	,	,	O	O
6	after	PREP	B–PP	O
7	unsuccessfully	ADV	B–VP	
8	seeking	V	I–VP	O
9	a	DET	B–NP	O
10	seat	N	I–NP	O
11	in	PREP	B–PP	O
12	the	DET	B–NP	O
13	united_states	N	I–NP	

Figure 3: Examples of the raw English data

The data was split 70-30 among the training and validation/test sets. There are a total of 53 relations in this dataset, but three were removed to keep the number of relations the same among datasets in hopes of a good comparison after the experiment. The data was originally formatted so that each sentence was tabularized, with the tokens being one column, the index (of the token) being another, the POS tag being the third, and finally a relation can be derived from the last two columns to another token in the sentence. For the data to be in the same format as the Chinese dataset, we restructure the English Dataset to have the same exact features and structure as the Chinese one. Therefore, the new examples look precisely like the ones in Figure 2, but in English. Note that since we removed the data from the full Chinese dataset, we now have the same number of training and test examples for each of the datasets. This allows for a more robust comparison.

5 Results

5.1 Original Model

This section of the experiment was split into two parts. The first was to train and test the model given a small subset of the data, and the second was to use the full dataset. To create small datasets, six relations were randomly selected from each dataset. For each relation, we selected 50 sentences from both the training sets and the test sets of both datasets that had the corresponding relation. The English dataset resulted in more examples for the subset, so we truncate the English data to only include the same number of sentences as the Chinese dataset. So, a total of 600 unique sentences were selected for each dataset.

We first trained the model with the small training set and tested it using the small test set. Finally, we compared the model's output with the true labels and computed the accuracy for each relation. The second part of this section of the experiment was similar to the first except the full dataset was used rather than a subset. Tables 3 and 4 show the results for both parts of this section.

Dataset	Subset Acc.	Subset Training Time
Chinese	0.1167	107min.
English	0.1583	102min.

Table 3: Accuracies for Chinese and English datasets (on subset) using the original model. The time column represents the time it took to train the original model on the subset.

Dataset	Full Acc.	Full Training Time
Chinese	0.3291	516min.
English	0.2936	499min.

Table 4: Accuracies for Chinese and English datasets (full dataset) using the original model. The time column represents the time it took to train the original model on the full dataset.

5.2 Updated Model

For this experiment, similar to section 5.1, we evaluate the model twice. However, this time, the updated model is used for predictions. As before, and in the same manner, 600 distinct sentences corresponding to 50 individual relations were selected from both datasets. The updated model was trained and tested using these subsets. Finally, the updated model was trained on both full datasets. Tables 5 and 6 show the results for both parts of this experiment.

Dataset	Subset Acc.	Subset Training Time
Chinese	0.1201	108min.
English	0.1588	100min.

Table 5: Accuracies for Chinese and English datasets (subsets) using the updated model. The time column represents the time it took to train the updated model on the subset of the data

Dataset	Full Acc.	Full Training Time
Chinese	0.3682	520min.
English	0.3345	512min.

Table 6: Accuracies for Chinese and English datasets (full dataset) using the updated model. The time column represents the time it took to train the updated model on the full dataset.

6 Discussions and Conclusions

6.1 Applications

Relation Extraction is the key component for building relation knowledge graphs, and it is of crucial significance to natural language processing applications. For example, it would be possible to extract the entire family-tree of a prominent personality using a resource like Wikipedia. In a way, relations describe the semantic relationships among the entities involved, which are useful for a better understanding of human language. In this section, two important applications of relation extraction are described: automatic question-answering and bio-text mining.

6.1.1 Question Answering (QA)

If a query to a search engine is "When was Barack Obama born?", then the expected answer would be "Barack Obama was born in 1961". The template of the answer is "born in" which is nothing but the relational triple PERSON, born in, YEAR. To extract the relational triples, a large database (ex. web) can be queried using a small initial question-answer set (ex: "Barack Obama 1961"). The best matching or the most confident patterns are then used to extract answer templates, which in turn can be used to extract new entities from the database. The new entities are again used to extract newer answer templates and so on till convergence.

6.1.2 Mining Biotext

Relation extraction methods are useful in discovering protein-protein interactions, and gene-binding conditions. Patterns such as "Protein X binds with Protein Y" are often found in biomedical texts where the protein names are entities which are held together by the "bind" relation. Such protein-protein interactions are useful for applications such as drug discovery. Cancer researchers can use inferences like "Gene X with mutation Y leads to malignancy Z" in order to isolate cancerous genes. These information patterns can be pieced together by extracting ternary relations between genes, mutations and malignancy conditions in a large corpus of biotext.

6.2 Conclusions & Future Work

We tested the original model for relation extraction and improved it by changing BERT_{base} to ALBERT_{Xlarge}. ALBERT_{xlarge} uses 44.4% fewer parameters than BERT_{base}, which effectively reduces overfitting. We see this in our results. Looking at Tables 3 and 4, we see that the original model achieves around 32.9% and 29.3% test accuracy on the full dataset for the Chinese and English datasets, respectively. From Tables 5 and 6, we see that the accuracy increases to 36.8% and 33.5% for the Chinese and English datasets respectively for the updated model. The training time and accuracy is about the same for both models for each dataset. This indicates that overfitting might have been an issue for the original model when creating the embeddings.

For future improvements, we might consider changing ALBERT_{xlarge} to $\text{ALBERT}_{xxlarge}$ or other variances of BERT. We can also add more instances to the dataset for better performance. Furthermore, the Conditional Random Field (CRF) layer in our model shown in Figure 1 could have been replaced by a CNN layer whose output will be fed to the next classifier in the model (Zhou et al., 2017).

7 Statements of Contribution

Peiyong Liu: Ran Chinese dataset experiments for models. Wrote Applications, Datasets and others.
Nikhil Krishna: Ran English dataset experiments for models. Created script to manipulate datasets. Wrote the Model and Results section as well as various other bits of report.
Niles Jiang: Modified scripts to change BERT_{base} to be ALBERT_{xlarge} to improve the model. Wrote part of the Abstract, Introduction and Related Work of report.

References

David Batista. 2020. Datasets of annotated semantic relationships. *GitHub*. Accessed: 2020-12-15.

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Systems with Applications*, 114:34–45.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Rohit J. Kate and Raymond J. Mooney. 2010. Joint entity and relation extraction using card-pyramid parsing. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning (CoNLL-2010)*, pages 203–212, Uppsala, Sweden.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations.

Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 402–412, Baltimore, Maryland. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality.

Makoto Miwa and Yutaka Sasaki. 2014. Modeling joint entity and relation extraction with table representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1858–1869, Doha, Qatar. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. Mobilebert: a compact task-agnostic bert for resource-limited devices.

Haoyu Wang, Ming Tan, Mo Yu, Shiyu Chang, Dakuo Wang, Kun Xu, Xiaoxiao Guo, and Saloni Potdar. 2019. Extracting multiple-relations in one-pass with pre-trained transformers. *CoRR*, abs/1902.01030.

Bishan Yang and Claire Cardie. 2013. Joint inference for fine-grained opinion extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1640–1649, Sofia, Bulgaria. Association for Computational Linguistics.

Peng Zhou, Suncong Zheng, Jiaming Xu, Zhenyu Qi, Hongyun Bao, and Bo Xu. 2017. Joint extraction of multiple relations and entities by using a hybrid neural network. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 135–146, Cham. Springer International Publishing.