## Knowledge extraction part：

### Task introduction：

In this part, we introduce the model we use to extract knowledge from **unstructured** data. The information extraction task, or more precisely, the "triple" extraction task, has the following sample data:

{  
  "text": "九玄珠是在纵横中文网连载的一部小说，作者是龙马",  
  "spo\_list": [  
    ["九玄珠", "连载网站", "纵横中文网"],  
    ["九玄珠", "作者", "龙马"]  
  ]  
}

You input a sentence and output all the triples that it contains. Where the triplet is of the form (S, P, O), and its S is the subject, the main entity, which is a fragment in the query; O is an object, a guest entity, which is also a fragment in Query. And P is predicate, the relationship between two entities. In general, (S, P, O) can be understood as "P of S is O".

### Sample characteristics

Obviously, this is a "*one to many*" extraction + classification task. Through the manual observation of the sample situation, it is found that its characteristics are as follows:

1. S and O are not necessarily words separated by the word segmentation tool, so it is necessary to annotate query to extract the correct S and O. Considering that participles may cut the wrong boundary, they should be annotated with word-based input.
2. Most of the sampling results are in the form of "one S and many (P, O)". For example, "the leading actors of Wolf Warriors include Wu Jing and Yu Nan", then "(Wolf Warriors, leading actor, Wu Jing)" and "(Wolf Warriors, leading actor, Yu Nan)" should be extracted.
3. The result of sampling is "multiple S, one (P, O)" and even "multiple S, multiple (P, o)". For example, "The leading actor in Wolf Warriors 2 and Wolf Warriors 2 are all Wu Jing", so "(Wolf Warriors 2, the leading actor, Wu Jing)" should be extracted.
4. The same pair (S, O) may also correspond to multiple P's. For example, "The leading actor and the director of Wolf Warriors are Both Wu Jing", then "(Wolf Warriors, the leading actor, Wu Jing)" and "(Wolf Warriors, the director, Wu Jing)" should be drawn.
5. In extreme cases, there may be overlaps between S and O. For example, "Autobiography of Lu Xun published by Jiangsu Literature and Art Publishing House", strictly speaking, in addition to "(Autobiography of Lu Xun, publishing house, Jiangsu Literature and Art Publishing House)", "(Autobiography of Lu Xun, author, Lu Xun)" should be extracted.

### Model description

The model is based on the method of "semi-pointer - semi-annotation" (removing CRF and replacing it with "0/1 annotation" to separate the structure of the starting and ending positions of the answers). The order is to extract S first and then pass in S to extract O and P. BERT [1](**B**idirectional **E**ncoder **R**epresentations from **T**ransformers) is adopted in the overall structure of the model：

1. After the original sequence is transferred to ID, Bert's encoder is passed in to obtain the encoding sequence;
2. The coding sequence is connected with two binary classifiers to predict S;
3. According to the passed S, the coding vector corresponding to the first and tail of S is extracted from the coding sequence;
4. Make a conditional Layer Norm of coding sequence by taking the coding vector of S as the condition;
5. The sequence after conditional Layer Norm is used to predict the Corresponding O and P of this S.

BERT's main idea was to construct a neural network for unsupervised training of large sample data, and then to adapt the model after training to other supervised training scenarios with various small samples by Fine Tunning method. In the past, neural networks trained a specific model from scratch for a specific scene, so to train a reliable model requires a large amount of data acquisition and manual annotation work.

The neural network trained by BERT's large sample is equivalent to providing a "generalized" pre-trained model. AI engineers can directly use this pre-trained model to solve domain specific problems through Fine Tunning.



Figure 1. Structure diagram of triplet extraction model based on Bert

When entity extraction is done with the structure of "semi-pointer - semi-annotation", it will face the problem of category imbalance, because generally speaking, the target entity words are much less than the non-target words, so the tag 1 will be much less than the tag 0. Conventional methods for dealing with imbalance can be used, such as Focal loss or manual adjustment of class weight, but after the use of these methods, it is not easy to set the threshold. I've done it in a way that I think is appropriate: I've raised the probability to the power of .

To be specific, I used to output a probability value of , representing the probability of category 1 which is , but now I change it into , that is, I think the probability of category 1 is . In addition, other than unchanged, loss still uses the normal binary cross entropy loss. Since we already have , so is going to be closer to , so the initial state is going to conform to the target distribution, so we can accelerate the convergence eventually.

The difference between the two can also be compared from the perspective of loss. Suppose the label is , then the original loss is：

And the loss to the *n* power is going to be the loss

Notice the , so when the label for 1, equivalent to amplify the loss of weight, and labeled , is closer to , so the corresponding loss smaller (also smaller gradient). Therefore, this is a kind of adaptive adjustment of loss weight (gradient weight).

Compared with Focal loss or manual adjustment of class weight, the advantage of this method is that without changing the original inner product distribution (is usually obtained by inner product plus SIGmoID), the distribution can be closer to the target, while without changing the inner product distribution is generally more optimization friendly.

**Experiment**

We use this model to extract triples as knowledge in the unstructured data. We use the DuEE Knowledge Extraction dataset to train our triple extraction model.DuIE Dataset is a large-scale human annotated dataset, with more than 450,000 SPO triples in over 210,000 real-world Chinese sentences, bounded by a pre-specified schema with 49 types of predicates. All sentences in DuIE dataset are extracted from Baidu Baike and Baidu News Feeds. DuIE dataset is to evaluate Schema based Knowledge Extraction algorithms.

There are two kinds of schema types: primitive types and entity types. Primitive types are for basic data types such as Number and Date. Entity types, also called classes, are the ones defined in schema representing types of entities, such as Person, Location, Organization, etc. In our dataset, subject type is always a class and object type can be either a class or a primitive type.

Table 1. Data Statistics

| DuIE Dataset | Total amount | Training set | Dev.set | Test set |
| --- | --- | --- | --- | --- |
| Sentence | 214,739 | 173,108 | 21,639 | 19,992 |
| Instance | 458,184 | 364,218 | 45,577 | 48,389 |

We train this model on a 2080Ti GPU for 20 epochs. The result of the model on test set is shown on table2. However, since we don’t have any dataset on military field, the real extraction result is lower than the experiment result. We craw military new from [www.sina.com](http://www.sina.com) and use the trained model to extract knowledge from the sentences. Totally, we get 8825 new entities and 17936 new triples from 20734 unstructured data to expand the knowledge graph.

Table 2. Evaluation result of our model

|  |  |  |
| --- | --- | --- |
| Precision | Recall | F1 |
| 0.65 | 0.71 | 0.68 |

[1] Devlin J, Chang M W, Lee K, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

[2] 苏剑林. (2019, Jun 03). 《基于DGCNN和概率图的轻量级信息抽取模型 》[Blog post]. Retrieved from https://kexue.fm/archives/6671

[3] 苏剑林. (2020, Jan 03). 《用bert4keras做三元组抽取 》[Blog post]. Retrieved from https://spaces.ac.cn/archives/7161