Universiteit Leiden

Master Computer Science

Named entity recognition on

Chinese biomedical patents

using pre-trained language models

Name:	Yuting Hu
Student ID:	2071932
Date:	15/01/2020
Specialisation:	Bioinformatics
1st supervisor:	Suzan Verberne
2nd supervisor:	Magnus Palmblad

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Recently, the high-speed advances in biological technology ensures a growing amount of biological studies, which offer more and more available biomedical text resources. Different text processing methods and models are then involved to mine information and knowledge from these resources, for example, Named Entity Recognition (NER), which is technology trying to recognize gene, protein, disease and other medical related entities, in biomedical domain. China has gradually come to play an important role in the global genomics-based testing and treatment market, leading to an increasing amount of Chinese biomedical text resources as well. However, there are only a few attempts to solve biomedical NER task on Chinese texts can be found during the past decade. Furthermore, as a new topic focusing on a very specific domain, there is even no previous attempt on Chinese biomedical patents NER task.

Thus, in our study, we built a possible solution to solve this extremely domain-specific biomedical NER problem. During our project, we built our own Chinese Biomedical patents dataset, then applied a BERT Pre-trained Language Model and several different learning methods, to let it understand Chinese text contents and solve NER task. Our optimal model finally archived a 0.54 ± 0.15 F1 score on our evaluation sets, then we did some further biomedical related analysis with generated predictions by the final trained model. These analysis indicates that our built solution and trained model is available to detect meaningful biomedical entities and novel gene-gene interactions, just with limited labeled data, training time and computing power.

Acknowledgements

First and foremost, I would like to thank my both supervisors, Suzan Verberne and Magnus Palmblad, for their generous help, useful advice and great supervision during this project. They always make me feel a lot warmth and inspirations during every talk and meeting with them.

Secondly, I would also like to thank my family for the support they provided me throughout my entire education process. Besides, I also want to thank my friend Zimu Wei, who offered help during our humanly annotation task, and Mohammed Charrout, who gave me a lot suggestions and helps on my study and life during the whole project period.

Last but not the least, I want to share my gratitude to all the creators and developers of all those wonderful open source projects and tools, which made our implementation possible and easier.

Contents

List of Figures

List of Tables

1 Introduction

In recent years, with the high-speed advances in biological technology, especially since the thrive of easier and cheaper next-generation sequencing, the whole-genome analysis has been progressed rapidly, which inspired and boosted biomedical research a lot. These growing amount of studies offer more and more available biomedical text resources, such as scientific literature and patents. In order to make better usage of these plentiful unstructured text data, different text mining strategies were integrated with biomedical area. Latest speedily evolving text processing methods and models ensure the opportunities for biomedical text mining to mine information and knowledge, then foster biomedical and drug discovery research in return[1].

In our study, we are about to seek a possible solution to solve a biomedical Named Entity Recognition (NER) problem, in order to recognize gene, protein and disease entities from Chinese biomedical patents data, which is English-Chinese code-mixing and has complex text writing style. In the rest part of this chapter, a detailed explanation of context and background knowledge from both text mining and biomedical aspect will be brought up, along with the introduction of several previous attempts in similar topics and descriptions of our motivation and research problem.

1.1 Biomedical Named Entity Recognition

Named entity recognition (NER) is a subtask of information extraction and text mining, that seeks to locate and classify named entity mentions in unstructured text into pre-defined categories. Compared with document or sentence level classification tasks, NER normally makes classifications on word or even character level, giving each word or character a category label that denotes whether it is part of a target named entity or not.

In biomedical text mining area, the application of NER is a widely discussed and studied topic, which aims to distinguish, for example genes, proteins, cell types or diseases, from the text in each document. These detected named entities will then be available for further statistic analysis or relation extraction task to offer evidence, resources or just give inspiration for biomedical research.

Naturally, once the concept of biomedical NER was first brought up, the need for well-organized and high-quality labeled dataset started to grow speedily. Several NER shared tasks have been built and organized to reach this need gradually. In 2003, the GENIA[2] corpus was collected by retrieving abstracts associated with

specific MEDLINE query terms such as "human", "blood cells" and "transcription factors". The release of the GENIA corpus promoted text-mining studies in the field of molecular biology and it serves as the seed for several tasks where truthful training and test sets can be constructed[3]. Then, in 2004, The JNLPBA shared task[4] is derived from five superclasses in the GENIA corpus while the entities are named protein, DNA, RNA, cell line and cell type, respectively.

1.2 Chinese Biomedical Patents

While most biomedical shared tasks and NER collections were organized by using either online articles or academic literature, as we mentioned above, patents are also huge and growing text resources available for biomedical text mining. Compared with other biomedical text formats, patents tend to contain some observations derived from or available for directly contributions in industrial areas, which probably would not appear in literature articles then. Thus, finding a proper way to analyse text contents and detect latest discoveries in patents has gathered interests and focus in order to give insights and inspirations for both companies and researchers.

In 2013, the Supreme Court of the US invalidated the company *Myriad Genetics*' claims to isolated genes, holding that, isolating genes found in nature are not patentable, which would definitely influence the biomedical patent publishing situation in the US during the following decades. Similarly, in 2015, Australian judges ruled that an isolated gene was a discovery rather than a patentable invention. Although Myriad's gene patents are still valid in many other countries, commentators anticipate that judges in those jurisdictions might follow the precedence of these suits and disallow the patentability of human genes[5].

Meanwhile, China has gradually come to play an important role in the global genomics-based testing and treatment market, leading to an increasing amount of biomedical discoveries in China as well. Moreover, according to the latest *Guidelines for Patent Examination (Guidelines)* issued by the State Intellectual Property Office of the People's Republic of China in 2010, isolated genes with an identified practical application are patentable in China then. In this case, whether there will be a lot of interesting biomedical discoveries patented in China but not in the US, especially during the period when that US patenting rule is valid, naturally became an interesting question into researchers' view.

1.3 Motivations and Problem Statement

We have discussed above that how important and helpful to apply text mining techniques on biomedical area, and why NER is one of the vital tasks here in biomedical text mining domain. If focusing on biomedical NER attempts globally, there has been a lot of work before solving this task with English text. However, when turning our attention to Chinese biomedical NER, it was a pity that only a few attempts can be found during the past decade. Detailed explanation and descriptions of all these attempts will be givin in section 2 Background. Furthermore, as a pretty new topic focusing on a very specific domain, there is even no previous attempt on Chinese biomedical patents NER, which means that we not only lack a proven well-performed algorithm or solution, but also failed to find any well-built benchmark dataset or even just available dataset.

Since the growing amount of Chinese biomedical patents text data, to solve the interesting assumptions we have currently related to gene patented situation in China, it is important to find a solution to process and understand these domain specific data. Thus, here we solve a NER task on Chinese biomedical patents text data, which is code-switched and have abundant uses of complex terms, in order to do further analysis focusing on situation of genetic discoveries patented in China.

To have a detailed resolving of this problem, we will need to build a dataset ourselves (both unlabeled and labeled) since our work can be considered as the first attempt on this specific topic. We will select a Natural Language Processing (NLP) model which can understand Chinese text contents and solve NER task. After we select the model and have our built labeled dataset, we train the model and generate predictions on the unlabeled dataset with our trained model. Finally we still need to do some post analysis on these generated predictions to mine useful information and meaningful insights in biomedical domain.

1.4 Summary and Thesis structure

In this section, we introduced the definition and situation of biomedical NER task, along with descriptions of the specialty of Chinese biomedical patents data and lack of previous NER attempt on this domain-specific data. Based on the challenges and problems we discussed above, after our final implementation and experiments, we can give a summary of the main contributions of our study:

- First attempt trying to solve this specific Chinese biomedical patents NER problem with limited labeled data;
- Built 2 large unlabeled Chinese biomedical patents dataset;
- Built a humanly annotated gold standard labeled dataset, which contains 5,813 sentences and 2,267 unique named entities from 21 patents;
- Built a NER classifier which is available to detect gene, protein and disease names in Chinese biomedical patents text using a BERT pre-trained language model;
- Further analysis focusing on Chinese biomedical discoveries patenting situations in recent years with the NER results generated by our built classifier and dataset.

In the following sections, introduction of previous work and related knowledge in both NLP and biomedical domain will be given in section 2 Background; our complete workflow, implementation details and some information of our built datasets will be explained in section 3 Methods; technical details and results of both benchmark and training experiments of our selected NLP models, along with the post analysis results, will be described and discussed in section 4 Experiments and Results; finally, in section 5 Conclusions and Future Work, the summary and limitation of our study will be given, which will then lead to discussions on possible future work.

2 Background

Before we explain our built dataset and learning methods, we first need to know some background knowledge of models or theories we will be using in our workflow. In this section, we will introduce and give original sources of both the important deep learning models and some bioinformatic concepts we applied further in building our data set and NER classifiers.

2.1 Text style and characteristics

Before we start to make use of these patents documents, we can imagine that it is definitely not a simple task to process Chinese biomedical patents data, because of some typical traits of its text style. Similar to other Chinese biomedical text, Chinese biomedical patents will have code-mixing or code-switching text which mainly because the protein and gene names are commonly written in English (or the English names been noted after the Chinese one), while the disease names and other contents will be written in Chinese. Moreover, even just inside each single named entity, it is possible that the code-mixed expression still appears. For example, possible formats of the same protein 'Interferon gamma' (official abbreviation form as 'IFN-gamma' or 'IFNG') in Chinese patents will be: 干扰素伽玛(pure Chinese format), 干扰素γ(Chinese short format), 干扰素-gamma(code-mixing format), IFN-γ(globally commonly used abbreviation format) or Interferon gamma(the original English name).

Besides, another vital problem of processing any Chinese text is that, if the source was PDF files or text converted by PDF files using optical character recognition (OCR), then, because of the large variety of Chinese characters and their complex shapes, the OCRed text will have relatively low quality compared with OCRed English text. If the text contents are in general domain, there has been several well developed tools which can detect or correct possible OCR errors automatically. But for domain specific text, popular existed tools based on simple rules can not handle the OCR error well, especially if there are a lot complex and difficult terminology using cases [6][7]. Facing this problem, we either need to build an OCR or OCR correction tool by training a complex model with our own domain specific dataset since we failed to find an existing one[8], or as what we decided to do, as a 'prototype' attempt, continue with current text to see what we can get in the further experiment. Then some improvements or future work can be extended by other researchers who have interests in this topic.

2.2 Prior work

2.2.1 Global biomedical NER models

If focusing on biomedical NER attempts globally, there has been a lot awesome work before solving this task with English text. [9] is a comparison study on several different relatively traditional text mining methods. It compared the NER classification results of 4 types of Word Embeddings combined with the Conditional Random Field (CRF), 2 simple Reccurent Neural Networks (RNN) and some statistical models such as Hidden Markov Model (HMM) and Support Vector Machine (SVM). It shows that their built system using Word2vec+CRF acchived the best performance, which was 72.82% F1 score on the JNLPBA dataset.

Since the rapid advancement of deep neural networks these few years and the amazing results archived by different Long short-term memory (LSTM) networks when solving different text mining tasks, more and more recent biomedical NER studies tend to use LSTM or other deep models as well. [10][11][12] are all different work trying to solve biomedical NER task using an architecture that contains a Bi-directional LSTM (Bi-LSTM) combined with a CRF final layer. [13] was an attempt to find whether it is possible to improve the performance when combine the Convolutional Neural Network (CNN) with the popular Bi-LSTM.

2.2.2 Chinese biomedical NER models

However, when turning our attention to Chinese biomedical NER, it was a pity that only a few attempts can be found during the past decade. [14] was the first work we can find related to this topic and the author believed that they did this Chinese biomedical NER task as the first person. It created and calculated several feature groups from its small gold dataset built with Chinese biomedical research abstracts, then applied a CRF model on it. Although they showed their best model got a 68.60±4.93(%) F1 score among 50 runs, since their dataset was small (481 sentences and 1062 entities in total), the train-test split may have influenced the results. Another sceptical point was that they split their train-test set by randomly selecting sentences from their whole corpus, which seems not correct since the dataset was actually built and organized in document level and this action may probably cause one document being separated then appear in both train and test set, which may definitely cause a little over-fitting in this case.

A recent attempt was [15], doing open concepts extraction from 4,931 biomedical articles which contain 41,733 sentences and 97,373 entities in total. The NER part did not specify different NE categories, just try to find all NEs. Besides, it only used dictionary matching and rule-based method for NE, the main classification model was for relation extraction. They got a 0.7604 F1 score for their non-category-specified open concept NER and finally reached 0.522 F1 score for their main relation extraction task.

2.2.3 General domain Chinese NER models

Since the huge differences between English and Chinese language, even both in the biomedical domain, it is still barely possible to take those English biomedical NER attempts as references or standards when we plan to build our Chinese biomedical patents NER system. Then naturally, we can take an eye on some latest work on general domain Chinese NER tasks. The state of art NER model at that time when we started our study was the Chinese Lattice LSTM[16], which applied the Lattice structured RNNs framework and made some novel modifications to ensure the original Lattics network possible to solve segmentation-free Chinese NER task.

In 2018, an inspirational work published by the Google research team was their attempt to find a way using transformers to do language understanding, the so called BERT model[17]. The high performance of BERT when applied on almost all downstream NLP tasks and the possibility to fine-tune its original pre-trained language model (LM) with little efforts and costs, made every researcher and company interested in finding solutions with BERT to solve NLP problems we could not solve before or just to improve current performance. There has definitely been attempts to solve Chinese NER in different domains with BERT as well. For example, [18] applied BERT on the Microsoft Asian Research Literature NER benchmark dataset (MSRA)[19], while [20] tried to solve the Chinese clinical NER and relation extraction task using BERT. All of them got promising results after fine-tuning the classifier within a relatively short time.

2.3 Pre-trained Language Models

Text mining has many advances during the past decade. One of the always vital text mining tasks is finding a representation of the un-structured text data, to ensure the computer can understand and use it. Previous attempts started by either using different encoding methods to encode text as numbers, or applying some statistical models on text to calculate the number representations[21]. As mentioned in the last subsection, deep neural networks has gained massive progress within these few years, which naturally inspired applications of complex and deep neural networks on text representation task[22]. Either those relatively traditional statistical models, or recent deep neural network models, as long as it aims to learn a representation of the text, it can be called a language model.

A language model trained on a sufficient amount of data containing general domain text can then be applied on a lot different downstream NLP tasks since it has learnt the representation of, or we can say 'understand', that language. In this case, this language model can be called a pre-trained language model. Nowadays, this transfer learning style application has become really popular and ensures a lot NLP tasks which cannot be solved or poorly solved before, more possibility to be solved today, by using existed well-performed language models. This can be implemented by either adding downstream task related layers on the

language model or just simply change the output layer. When need to solve NER tasks in specific domain, we can still apply the pre-trained language model which were actually trained on general domain data, just by fine-tuning the pre-trained model with domain-specific data.

2.4 NER Evaluation Metrics

The performance of classification problem is typically evaluated by a confusion matrix as illustrated in Table 2.1 (for a 2 class problem). Here the 'positive' or 'negative' depends on the class or label we are interested in. When coping with text data, or we can say in NLP domain, precision, recall and F1 score are usually applied as evaluation measures, which can be calculated as:

- Precision = TP/(TP+FP);
- Recall = TP/(TP+FN);
- F1 = 2*(Precision*Recall)/(Precision+Recall).

Table 2.1: Confusion matrix

	Predicted Negative	**Predicted Positive**
Actual Negative	TN	FP
Actual Positive	FN	TP

However, for NER task, which assigns a classification label for each word or character in one sequence(sentence), naturally comes the problem that whether to calculate the confusion matrix in word level or named entity level. Here we applied the measurement method introduced at the conference CoNLL-2003, which measures the performance of the systems in terms of precision, recall and f1-score, where:

- Precision is the percentage of named entities found by the learning system that are correct;
- Recall is the percentage of named entities present in the corpus that are found by the system;
- A named entity is correct only if it is an exact match of the corresponding entity in the data file[23].

The F1 score has been mentioned many times when we introduced the performance of prior work in section 2.2. In our study, we will also mainly focus on F1 score measurement when we explain and discuss the results of NER, while precision and recall scores will still be calculated and given in detailed results table then.

2.5 Biomedical Datesets and Databases

Because of the high-speed advances in bioinformatics, nowadays tons of biological datasets and databases are available for researchers in either biological or data science areas. For example, as mentioned before, several biomedical NER datasets, like JNLPBA and GENIA corpus, helped and inspired a lot further studies. Similarly, BC2GM is a biomedical NER dataset for BioCreative II Gene Mention detection shared task, which contains 15000 training and 5000 test sentences derived from PubMed abstracts, with humanly annotated gene mentions[14]. Besides, collaborations between different universities and research institutions offer more and more well-built and huge in size genetic databases to the world. HGNC is a database which stores unique symbols and names for human locus[24], while the Universal Protein Resource (UniProt) is a comprehensive resource for protein sequence and annotation data[25].

For the general medical domain, since the growing needs for hospitals and medical researchers to store precious real-world clinical data in a structured way, the International Classification of Diseases (ICD) has been built and maintained for a long time by World Health Organization[26]. ICD-11 is the 11th version of ICD, which is the diagnostic classification standard for all clinical and research purposes. With these biological and medical dataset and databases, we can collect and extract approved gene, protein and disease names, then use then to build dictionaries for automatic annotation matching, and also for our biomedical related post analysis after experiments.

String is a database of known and predicted protein-protein interactions. The interactions stem from computational prediction, from knowledge transfer between organisms, and from interactions aggregated from other (primary) databases[27]. In our study, we will build gene-gene connection network with predictions generated by our trained model. Then we can apply the existed network in String database, to compare and analysis our new generated networks. Detailed methods how we applied these datasets and databases will be described in section 3 Methods.

3 Methods

3.1 Data collection and dataset building

Since our study can be considered as the first attempt to do NER with this very specific domain text: Chinese biomedical patents, which means that we not only lack experiments data to train our model, but also can not use general domain data as alternative. Thus, the first step of the whole work will be collecting data and start to build our own dataset.

3.1.1 Collecting patents

To collect large amount of in domain patents data at first stage, we need to select a proper source or we can say patents database which should allow us to do customized searching, it should also have updated and abundant patents available. We did some simple analysis on the Google Patents[28] and Chinese official CNIPA[29] patents search engine. We checked the patents searching results with the same keywords on both website, then found that, Google Patents searching results covered all results returned by CNIPA and can even return more results than CNIPA some years, although this were probably caused by different versions of one same patents. Besides, Google Patents offers both plain text source in its HTML page and PDF file source while CNIPA only offers PDF source. Based on these results, we finally decided to use Google Patents database and wrote our scripts to automatically download all available patents files with specific searching keywords. Here we collected two groups of Chinese patents, one with searching keyword "人类AND基因" which means "human AND gene". We retrieved patents within 1st January 2009 to 1st January 2019 with patent code starting with 'CN'. The other with keyword "乳腺癌AND生物标记物" which means "breast cancer AND biomarker" within 1st December 2012 to 1st January 2019 and with patent code starting with 'CN' as well (we will call them 'HG' and 'BC' to refer to these 2 dataset in the following).

Then we implemented a patent code filter to improve the real relatedness of the searched patents to the keywords. The International Patent Classification (IPC) provides for a hierarchical system of language independent symbols for the classification of patents and utility models according to the different areas of technology to which they pertain[30]. We first did some analysis on the distribution of the IPC codes of all patents in the same group, then based on the analysis, we set the IPC codes we need to keep or filter. The detailed IPC codes we kept or filtered for each group of patents are shown in the Appendix.

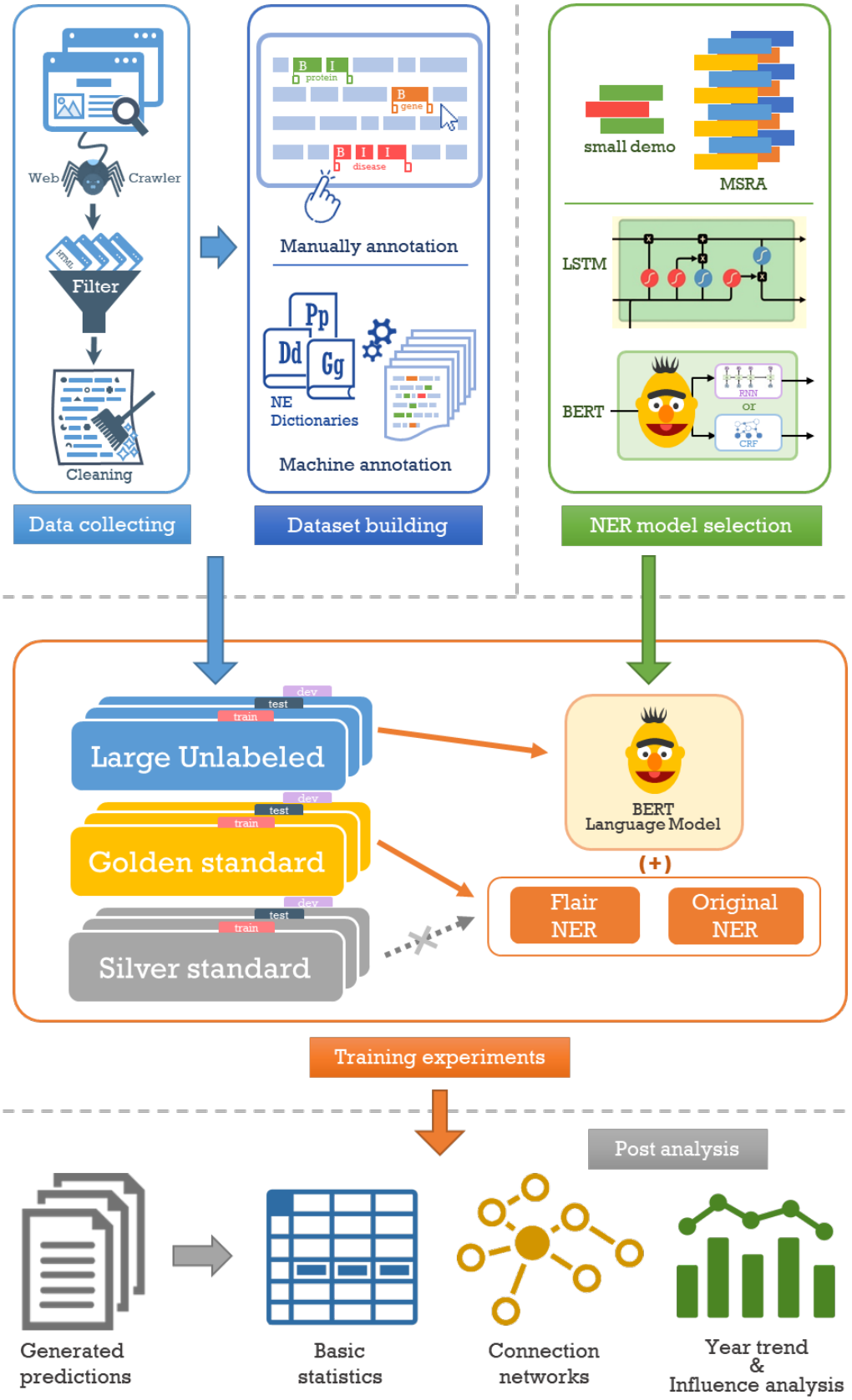


Figure 3.1: Overall workflow diagram

Text cleaning
After we applied our IPC code filter, we had 2659 and 53007 patents in the BC and HG datasets, respectively. As we described in section 1 Introduction, although we extracted these patents from the searched HTML webpage, the patents text stored in the HTML seems also been converted from its original pdf files since we noticed the differences from the HTML text and the pdf file availble on the same webpage results and we can detect that thoes differences are mainly some OCR errors. Although it is not available to use some general Chinese text correction tools since the huge differences between biomedical patents text and general domain text, we can still apply some basic text cleaning to improve the text quality before we start annotation. These including:

1. Removing blank lines and redundant spaces inside each line;
2. Replacing space inside English and code-mixing parts with '-' symbol.

The first 2 steps is mainly led by the typical Chinese language processing style, which is processing text in character level instead of word level. Thus, the splitting of sentences was then purely implemented by splitting by lines. In the final BIO annotation files, each line contains only one character followed by its corresponding BIO tag, and each space or blank line in the file denotes the separation of sentences. Thus, normal spaces appearing in English or code-mixing parts will cause incorrect sentence splitting problem when feeding the text file to classifiers or doing further post-processing.

3.1.2 Human annotated dataset

After we got our cleaned unlabeled patents data, we decided to build a small human manually annotated dataset as our gold standard set, which would then work for:

1. Training and evaluating the downstream NER task layer added on both pre-trained and our re-trained language model;
2. Evaluating the quality of further silver standard dataset.

We randomly selected 21 patents from the total collection of our two unlabeled dataset, then only annotated gene, protein and disease named entities. The named entity appeared in the patent text would be annotated with BIO format tags in character level.

We tried 2 different open-source annotation tools, which are doccano[31] and YEDDA[32]. While doccano is well-built with user-friendly UI and UX design and can be deployed online, YEDDA looks a bit old and not very user friendly and does not offer any online deployment function. However, one vital reason we finally applied YEDDA as our mainly annotation tool was that it offers the automatic annotation function which simply just detecting named entities based on current existed annotations in the same document. This function helped our annotation work and saved our time a lot since the specialty of biomedical patents writing style that one term will be used and mentioned repetitively very often in

the same document. Besides, it also offers a native inter-rater analysis function which may also be useful if needed.

Rules and standards

We set up some standards and rules for our annotation work. The general annotation standards and rules are:

- 2 raters/annotators (inter-rater check);
- Read whole patent content to understand reference and meaning of the terms first;
- In each paragraph or sentence, first need to read and understand the contents and contexts well, to make sure whether one specific term mention refers to the gene or protein (e.g. the annotator should understand and be clear that the mention "the gene which coding the protein X" refers to a gene but not protein);
- No nested or overlapping named entities allowed;
- Annotate the longest meaningful named entity preferentially;
- If detect a spelling or OCR error, still annotate the named entity when the annotator make sure that there existed one inside its corrected format.

For each named entity type, we also set detailed rules on how to recognize and categorize them. The named entity should be annotated as **protein** in these cases:

- Growth factor (e.g. cytokine and other signal proteins);
- Most enzymes, enzyme families, or one category of special enzymes (e.g. DNA polymerase, Acetyltransferase) except RNA enzymes (e.g. Ribozyme);
- Most antibiotics;
- Protein family (e.g. Histone, tubulin);
- Protein expression of a specific gene;
- Antibiotic drug conjugate (e.g. ADC);
- Peptide(s) or amino acid(s);
- Part of protein structure.

The named entity should be annotated as **gene** in these cases:

- A protein coding gene;
- Primer;
- Nucleotide(s), nucleotide analogs (e.g. Adenine, 5-propanepyrimidine);
- Ribozyme;
- Gene probes, DNA microarrays and other gene products;
- Gene family (e.g. DNA damage repair genes);
- Expression vector: plasmid, specific ones constructed and named by the patent holder;

The named entity should be annotated as **disease** in these cases:

- Traditional disease names or symptom (e.g. headache, stomachache);
- Formal disease names (e.g. B-cell lymphoma, breast cancer);
- A disease with some specific resistance;
- Early or advanced period of disease;
- Tumor or cancer molecular subclass (e.g. 三阴型/triple negative breast cancer, 胃样癌/gastric-like).

Built dataset information

Now, we have 2 large unlabeled dataset and one small gold standard humanly annotated dataset, here we can do some basic statistics then show the information of our 3 built dataset. The text contents information of all these 3 dataset is shown in Table 3.1.

Table 3.1: Document information of built dataset

dataset: large unlabeled					
	n_docs	**n_sents**	**n_chars**	**avg_doc_length (sents)**	**avg_sent_length (chars)**
BC	2,659	1.08M	160M	405	150
HG	53,007	21.75M	2.84B	410	130
dataset: manually annotated					
gold_set	21	5,813	0.78M	277	134

The annotation information of the gold standard set is shown in Table 3.2. The row names indicate whether the value in table shows the number of total appearance or only unique appearance. The column names indicate the each single category, while 'all' indicate all 3 types of annotations.

Table 3.2: Annotation information of manually annotation dataset

	gene	**protein**	**disease**	**all**
total	1888	5030	2739	9657
unique	482	1053	732	2267

3.1.3 Silver standard dataset

Since we have built two large unlabeled dataset, besides making a relatively small humanly annotated set, we also want to seek a way to make fully use of the rest huge amount of data. One idea is to build a silver standard or here we can say distantly-supervised dataset, which is annotated by machines or algorithms with existed humanly made annotations. To make sure the sufficient amount and decent quality of the source or reference annotations, one commonly applied method is to collecting annotations from existed well built gold standard dataset.

Version 1 gene and protein names
Since one important code-mixing trait of Chinese biomedical patents is that, most protein and gene names are written in English while the disease names and rest contents are in Chinese, here we decided to collect and extract English protein and gene annotations from JNLPBA[4] and BC2GM[14] dataset. More detailed, we only extracted **gene** annotations from BC2GM while both **gene** and **protein** annotations from JNLPBA. After collecting, these extracted annotations still need cleaning and organizing. We first removed all redundant appearances make sure each appearance of gene and protein in the final annotation list is unique. Then we removed all single letter and only-digital annotations since these would cause a lot meaningless matching in the patents text.

Version 2 gene and protein names
Besides using the existing well-built English Bio-NER dataset,in order to improve the reliability and authenticity of our source gene and protein annotations, we also decided to apply existed bioinformatics databases. We make a query in HGNC[24] to search for **genes** still academically approved so far, and in UniProt[25] database to get only human **proteins** which are manually reviewed. Both searching results can be directly downloaded and then ready to use after doing some organizing. These group of reference gene and protein annotations were considered as the 2nd version source while the previously extracted group from Bio-NER dataset as 1st version source.

Chinese disease names
Then, we extracted Chinese **disease** names from the officially Chinese translated version of ICD-11[33]. We only retained meaningful categories here, which should meet these requirements:

1. Should be one of the leaf nodes which do not contain any child node if considering the ICD system as a tree structure;
2. Should be a disease name despite of drug names, physical information of diagnose and other non-disease categories;

Based on these 2 rules, we only retrieved these categories from ICD-11: A-L, N, S and XH. Then we also removed all single character disease names to avoid a lot meaningless matching in further experiments.

Now we have our 2 versions annotation collections ready, we then applied the Aho–Corasick algorithm, which is a string-searching algorithm can solve matching task with huge reference 'dictionary' with only linear complexity[34]. Unfortunately, after using the gold standard set to evaluate our generated annotations which match our extracted 'dictionary' on the unlabeled patent data, the quality of this silver standard set was too poor to be applied on any further experiment. So we **would not** consider to use this set during the design and implementation of the training and predicting experiments then. The evaluation method and results of this silver standard set will be given in section 3.3 Evaluation.

3.2 Models and learning methods

When we were doing some work with gathering and organizing our data, we started doing some researches and comparison of different Language models, which were popular and performed well in Chinese NER task, at the same time. We finally decide to do some benchmark comparison experiment to choose one model between LatticeLSTM and BERT language models, since they both were new and State-of-the-art models when it first released and both offered sufficient supports for other researchers to replicate or use their models. The detailed information of benchmark dataset we applied, along with the comparison experiments settings and results will be described in section 4 Experiments and Results.

After the benchmark experiments, we finally decided only use the BERT model to build our further training experiments, because it not only performed better than LatticeLSTM on both datasets, but also much more efficient especially on huge datasets. Based on the size and type of dataset we already built, which were 2 large unlabled and one small gold standard labeled, we implemented 3 different learning methods with BERT language model (diagram explanation shown in Figure 3.2).

- **Supervised original**: fine-tuning all weights (BERT model layers plus NER layer) using a relatively small learning rate, with our **gold standard dataset**;
- **LM mixed fine-tuning**: first directly tuning weights of the BERT language model layers with **unlabeled dataset**; then repeat the supervised original learning step;
- **PartBERT+CRF fine-tuning**: fine-tuning weights of part of the BERT model (last 4 layers) plus an added CRF layer, with our **gold standard dataset** (implemented with FlairNLP[35] package).

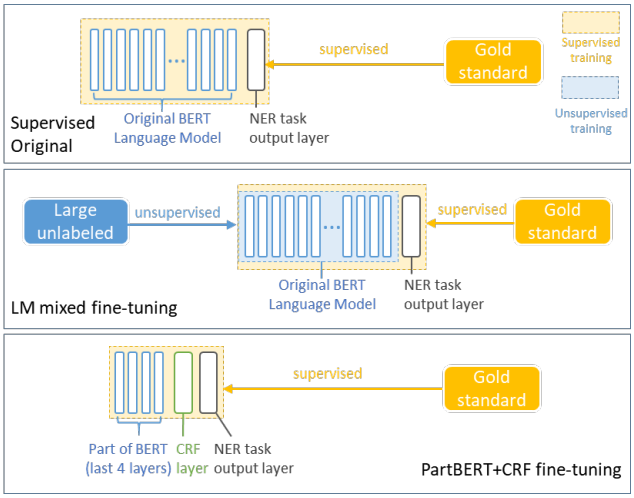


Figure 3.2: 3 different learning methods

3.3 Evaluation

3.3.1 Dataset quality

As mentioned in several paragraphs before, to monitor not only the quality of our building dataset, but also the performance of our built models and learning methods, we need to choose appropriate methods and measures to evaluate several parts of our experiments. These evaluations all rely on our humanly annotated dataset and the idea to consider it as ground truth (gold standard). Thus, it is important to monitor the quality of this dataset first.

For common annotation tasks, Cohen's Kappa is considered as the standard measure to calculate inter-rater argument. Since we had 2 annotators for part of our patents, we were supposed to monitor this measure to check the quality of our gold standard dataset. However, recent years, it has been proved that for NER tasks, Kappa does not seem to be the best measure[36]. This is because Kappa needs the number of negative cases, which isn't known for named entities. Thus, here we calculated the communal F1 score of the 5 patents between the 2 annotators. We got a 0.95 average F1 for this 5 files, and 0.98, 0.91 and 0.97 F1 score for gene, protein and disease type entities, respectively, which told us that it was good enough to be considered as our gold standard set then.

After we make sure our humanly annotated dataset is reliable enough, we can use it as gold standard set to analyze our silver standard set(automatic matching set). Here we treat this as a common classification evaluation task, that means, we can calculate the precision, recall and F1 scores measurement values of the overlap between the silver and gold standard set. Unfortunately, both our silver standard set using the 2 versions reference annotations respectively, only got less than 0.1 for all measurement scores, which was definitely not good enough to be used to train any of our model. Possible reasons of this results and future improvements of this silver standard set will be discussed on section 5 Conclusions and Future Work.

3.3.2 Model training performance

Based on some traits of Chinese biomedical patents text that both the writing style and usage of entities differs a lot in each different patent. Thus, during train-test set building step, if one patent was split into 2 parts which one part appears in train set while the other in test set, which would definitely lead to over-fitting. Thus, to avoid this problem, we should split train-test set on document level, which means, one patent can only appears in either train or test set and cannot appears in both.

Beside, as described in section 3.1.2 Human annotated dataset, our gold set contains 21 Chinese biomedical patents randomly selected from the total collection of our two unlabeled dataset. Although it has more then 5.8k sentences and 2.2k

unique named entity appearance in total, it is still a relatively small dataset. To make sure the model can learning enough information from the data, we decided to split the gold standard set to 18, 2 and 1 patent for train, test and dev set, respectively. Thus, we can imagine that, the 2 patents which were chosen as test set can influence the test evaluation results a lot, this guess was also proven by our real experiments results later.

To get more reliable evaluation results and monitor the stability of our models and learning methods, we decided to make 5 versions of the train-test-dev set groups, to implement a **cross-validation-like** evaluation method. Here, we randomly split the total set into 7 folds and each contains 3 patents. Then every time to form a evaluation set group (cbp set), select one fold by order, use 2 patents in the fold as test set and another 1 as dev set. The rest 18 patents (6 folds) would be considered as train set of that group. The process is also explained and shown in Figure 3.3.

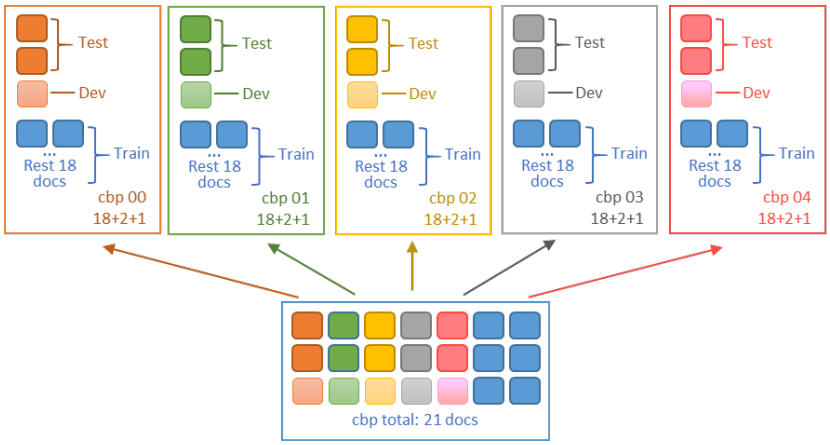


Figure 3.3: Split method of cross-validation-like evaluation sets

We call it cross-validation-like evaluation but not a cross-validation is because we did not build 7 datasets (evaluation set groups) which use each of the total 7 folds as the test(plus dev) set, due to the concern of further experiment efficiency, since every time one model will be trained and evaluated on all evaluation groups. Here we show the detailed information of these 5 set groups is in Appendices.

As described above, each cbp sets group contains 18, 2 and 1 patent in train, test and dev respectively. We will calculate the precision, recall and F1 scores for each category of named entity and the Micro and Macro averages of all mentioned measurement scores among all named entity categories. Then, the average and standard deviation among all 5 groups cbp sets can also be calculated and monitor both the stability and more truthful performance of all our implemented models and learning methods.

3.4 Post processing and analysis

3.4.1 Predictions generation and cleaning

Based on the evaluation results of the training experiments, we can then choose the optimal learning model, then apply the model on the full collection of our gold standard set to get our final trained model. This model would be applied on the huge unlabeled dataset to generate named entity predictions on the patent text. We finally generated biomedical entity predictions on all 2,659 patents in BC dataset and the 10,100 patents in part of HG dataset.

However, since the annotations are generated by predicting, it is possible that there are some meaningless named entities or the 'BIO' tags are not located in good order ('I' tag should always follows one 'B' tag and the type of 'I' tags can not be mixed). Thus, before we start some analysis, some post-processing cleaning steps were needed. We first implemented a detector to detect annotated named entities from the predictions. The rules to detect named entity are:

1. If one non-'O' sequence only contains one single character, discard it since we assume a single character named entity of either gene, protein or disease category will be meaningless;
2. Find any non-'O' sequence that starts with a B followed by only I-tags;
3. After step 2, if the tags belongs to the same category (gene, protein or disease), store it; else, only store the first part with 'BI...' tags belongs to the same category; other parts discarded.

Besides, after solving the 'BIO' tag location problem, we then filtered out the named entities which contain non-alphabetic characters except '-' or '_', to make sure all remained named entities are meaningful. With these cleaned named entities, we can do some post analysis to mine useful and meaningful information from our patents data.

3.4.2 Post statistics

We first did some basic statistics which analysis the overall NER information of the predictions, for example, number of unique and total mentions of all categories and several most commonly mentioned entities of each category. Then, we analysis how the entity mentioned in our predictions changed with time.
We also focused on several biological technologies, including Next Generation Sequencing(NGS), Whole Genome Sequencing(WGS), Polymerase Chain Reactionand(PCR), Electrospray Ionization(ESI), Capillary Gel Electrophoresis(CGE) and Sanger Sequencing, to analysis the differences of biomedical entity prediction situation among patents mentioned different biological technology. Here we set a synonyms list for all the technology, then did a simply matching to check whether one patent mentions any of the synonyms, then the patent would be

considered as using or mentioning corresponding technology. The detailed synonyms lists are shown in Appendices.

We also tried to build the gene-gene connection network using co-occurrence network concept. The idea is that, we assume all gene entities, which appear in the same 2 sentences in one patent, are all 'connected'. The node weights and edge weights would be updated and calculated as the number of patent documents which mention that node/edge. To get better visualization performance and more reliable predicted edges, we would set a threshold on edge weights, which equals the number of patents mentioning that edge. Any edge has weight smaller than the threshold will not be involved in the final network which will be visualized and analyzed then. Different thresholds were set to meet different requirements in order to solve different tasks better. Thus, the detailed edge weight threshold will be given in section 4 Experiments and Results when we describe and explain the post analysis results then.

4 Experiments and Results

In this section, some technical details of our implementation and experiments will be given, along with the final results and explanation of our results. All experiments and analysis were implemented with Python, and all benchmark, model training and final predictions generating experiments were run on one single NVIDIA Tesla K80 GPU with 11.5GB memory.

4.1 Benchmark experiments

We run some experiments using 2 benchmark datasets, to get a rough comparison between the model LatticeLSTM and BERT. The small demo is a sample NER dataset offered by the original author of LatticeLSTM under its project repository[37], while MSRA is a relatively big dataset, stands for the Microsoft Asian Research Literature NER dataset. MSRA is commonly applied in most STOA Chinese NER tasks and considered as benchmark dataset[19], while small demo is really small and limited but can ensure a quick start and check of the model performance on limited data. Detailed text contents and named eneity information of both dataset was shown in Table 4.1.

Here we run the LatticeLSTM NER with the original codes offered by its author[37]. Since the BERT model itself was not trained for solving NER tasks, some modifications were needed to run it then. We applied and modified the codes offered by [38], using the old-version official BERT PyTorch pre-trained model. Then we also applied the FlairNLP package to implement a part-of-BERT version, which was the same as the PartBERT+CRF model that we would use in the final training experiment and has been explained in subsection 3.2 Models and learning methods. The benchmark experiment results are shown in Table 4.2.

Table 4.1: Benchmark dataset information

dataset: small demo(total/unique)								
	n_sents	**n_chars**	**avg_sent_l**	**PER**	**ORG**	**GPE**	**LOC**	**all_entities**
train	1148	49732	43	495/266	441/251	1202/289	142/57	2280/863
test	316	14405	46	193/83	64/34	67/32	14/8	338/157
dev	113	5478	49	11/11	31/21	91/46	52/17	185/95
dataset: MSRA(total/unique)								
	n_sents	**n_chars**	**avg_sent_l**	**PER**	**ORG**	**LOC**	**all_entities**	
train	46364	2169876	47	17375/5918	20050/7887	33269/4662	70694/18467	
test	4365	172601	40	1413/635	1267/518	2641/542	5321/1695	

Table 4.2: Benchmark experiments results

dataset: small demo				
model name	**precision**	**recall**	**f1 score**	**training time (s/epoch)**
LatticeLSTM	0.5467	0.3382	0.4182	428
FlairBERT	**0.7686**	**0.7570**	**0.7628**	365
OriginalBERT	0.7292	0.6388	0.6910	**55**
dataset: MSRA				
model name	**precision**	**recall**	**f1 score**	**training time (s/epoch)**
LatticeLSTM	0.9196	0.9111	0.9153	26741
FlairBERT	**0.9319**	0.9165	**0.9241**	2779
OriginalBERT	0.9173	**0.9303**	0.9237	**2224**

We can notice that, LatticeLSTM got way worse performance than both BERT NER models on the limited small size demo dataset, while slightly worse then both BERT models on the big MSRA dataset as well. However, at this stage, the efficiency is also a very important measure since we had two huge dataset and several possible training experiments to be run within a limited time. It is obvious that although LatticeLSTM had good enough performance on the larger dataset, its running speed is too low for our further experiments, as shown in the results table. This was probably mainly because that the LatticeLSTM model is a RNN-based model, which complexity can increase sharply while the input sequence length increases since it need to 'remember' a lot memory or information of the former parts of one sequence.

Thus, we decided only building our experiments NER models with BERT then, and both these two implementations of BERT will be applied since they got similar performance on either efficiency or classification results.

4.2 Model training experiments

After we selected BERT as our only Language Model to be applied in building our training models, the BERT original developing team collaborated with hugging face team to release the updated new version of the PyTorch BERT implementation and pre-trained model, which ensure an easier way to implement directly (re)training on whole BERT Language Model then. Thus, we implemented the LM mixed fine-tuning model(described in subsection 3.2) using the hugging face BERT implementation[39] in the following experiments, while the other 2 models were implemented in the same way as in benchmark experiments.

Before running our experiments, we still need some preparation for the Language model training. Although BERT can directly use unlabeled text data to train its Language model, to make sure the efficiency as well, we split 2 groups of smaller sets out our 2 large unlabeled dataset. Both the smallBC and partHG groups

Table 4.3: Dataset for fine-tuning language model

	n_docs	**n_sents**	**n_chars**	**avg_doc_length (unit: sents)**	**avg_sent_length (unit: chars)**
smallBC_train	100	41,016	6.13M	410	149
smallBC_test	10	2060	0.35M	206	172
partHG_train	10000	4.18M	0.54B	418	130
partHG_test	100	36,097	5.15M	361	143

contains a train and test set and would be finally used to train the Language model, detailed information of these 2 groups are shown in Table 4.3.

Now we can run our training experiments on all 3 models. For the final NER layer, each model was trained for 40 epochs with the train set and a small dev set, then evaluated on the test set, among all 5 cbp dataset (described in subsection 3.3 and detailed information shown in Table A.1).

The final results are shown in Table 4.4. Here we only show the average F1 score and standard deviation of each model among our all 5 cbp dataset. The row name 'PartBERT+CRF', 'Supervised Original' and 'BERT LM mixed' represent our 3 models or training methods described in subsection 3.2 Models and learning methods.

The optimal results of each category has been noted as bold in the table. We can notice that, the BERT LM mixed model which was trained on partHG dataset for only 1 epoch got most optimal results in all categories, while actually all models or training methods did not differ a lot. It seems like, indeed more data and more training steps (or training epochs) can lead to better performance, this was also indicated in original BERT paper[17]. However, the results also showed that, the more data and longer training method can cause less stability if focusing on the standard deviation shown in the table, fortunately they did not differ too much among different models as well.

Table 4.4: Training experiments results

average_f1 (among_5_dataset)	**gene**	**protein**	**disease**	**macro avg**	**micro avg**
PartBERT+CRF	0.21 ± 0.08	0.25 ± 0.20	0.67 ± 0.12	0.38 ± 0.03	0.47 ± 0.11
Supervised Original	$\mathbf{0.31 \pm 0.21}$	0.34 ± 0.11	0.60 ± 0.09	0.42 ± 0.09	0.49 ± 0.16
BERT LM mixed (smallBC_1epoch)	0.21 ± 0.18	0.33 ± 0.23	0.67 ± 0.16	0.40 ± 0.06	0.51 ± 0.15
BERT LM mixed (smallBC_30epochs)*	0.26 ± 0.19	$\mathbf{0.36 \pm 0.24}$	0.67 ± 0.12	$\mathbf{0.43 \pm 0.06}$	0.52 ± 0.15
BERT LM mixed (partHG_1epoch)	0.27 ± 0.21	0.33 ± 0.22	$\mathbf{0.70 \pm 0.12}$	$\mathbf{0.43 \pm 0.06}$	$\mathbf{0.54 \pm 0.15}$

* The 'smallBC' and 'partHG' indicates which unlabeled dataset the Language model was trained on, while the '1epoch' and '30epoch' denotes the number of epochs the language model was trained for.

Another point is that, while all category, plus the two average scores did not got very high results, all models achieved higher performance on recognizing 'disease' category entities. This was probably mainly because most disease names in Chinese biomedical patents dataset were written in Chinese without any code-mixing situation, and the original pre-trained BERT model we applied was a general domain Chinese language model as well. It seemed reasonable that our trained model can solve recognizing pure Chinese named entities better than code-mixing cases then. Detailed classification results including all precision, recall and F1 scores of each model on each cbp dataset are shown in Appendices.

4.3 Post analysis

4.3.1 Basic information

After we generated some predictions on our unlabeled datset, we applied some post processing steps to clean the predictions. The basic information of our cleaned predictions on BC and HG dataset are shown in Table 4.5 and Table 4.6, respectively. The **gene**, **protein** and **disease** entities are placed in **green**, **blue** and **light orange** cells, respectively.

Table 4.5: Predicted named entities information of dataset BC

	gene	**protein**	**disease**	**all**
total	410,523	933,106	548,871	1,892,500
unique	70,026	129,791	45,047	244,864
top10	HER2	单克隆抗体 (Monoclonal antibodies)	乳腺癌 (Breast cancer)	乳腺癌
	VEGFR2	半胱氨酸 (Cysteine)	肺癌 (Lung cancer)	肺癌
	EGFR	抗体片段 (Antibody fragment)	前列腺癌 (Prostate cancer)	前列腺癌
	VEGFA	EGFR	卵巢癌 (Ovarian cancer)	单克隆抗体
	KRAS	贝伐单抗 (Bevacizumab)	胰腺癌 (Pancreatic cancer)	卵巢癌
	CDR3	双特异性抗体 (Bispecific antibody)	胃癌 (Gastric cancer)	胰腺癌
	c-MAF基因 (c-MAF gene)	HER2	肝癌 (Liver cancer)	胃癌
	PLGF	轻链可变区 (Light chain variable region)	结肠癌 (Colon cancer)	半胱氨酸
	CDR2	重链可变区 (Heavy chain variable region)	膀胱癌 (Bladder Cancer)	肝癌
	FGFR3	VEGF	白血病 (leukemia)	结肠癌

Table 4.6: Predicted named entities information of dataset HG

	gene	protein	disease	all
total	258,572	728,815	319,571	1,306,958
unique	186,601	442,791	187,674	817,066
top10	dsRNA	单克隆抗体 (Monoclonal antibodies)	乳腺癌 (Breast cancer)	单克隆抗体
	CXCR4	赖氨酸 (Lysine)	糖尿病 (Diabetes)	糖尿病
	胞嘧啶 (Cytosine)	IgG	肺癌 (Lung cancer)	赖氨酸
	核酶 (Ribozyme)	重链可变区 (Heavy chain variable region)	前列腺癌 (Prostate cancer)	IgG
	TRPA1	轻链可变区 (Light chain variable region)	卵巢癌 (Ovarian cancer)	重链可变区
	scFv	半胱氨酸 (Cysteine)	结肠癌 (Colon cancer)	轻链可变区
	siRNA	免疫球蛋白 (Immunoglobulin)	胰腺癌 (Pancreatic cancer)	肺癌
	hTERT基因 (hTERT gene)	CDR	胃癌 (Gastric cancer)	半胱氨酸
	HER2	抗体片段 (Antibody fragment)	类风湿性关节炎 (Rheumatoid arthritis)	前列腺癌
	利巴韦林 (Ribavirin)	蛋白酶 (Protease)	动脉粥样硬化 (Atherosclerosis)	免疫球蛋白

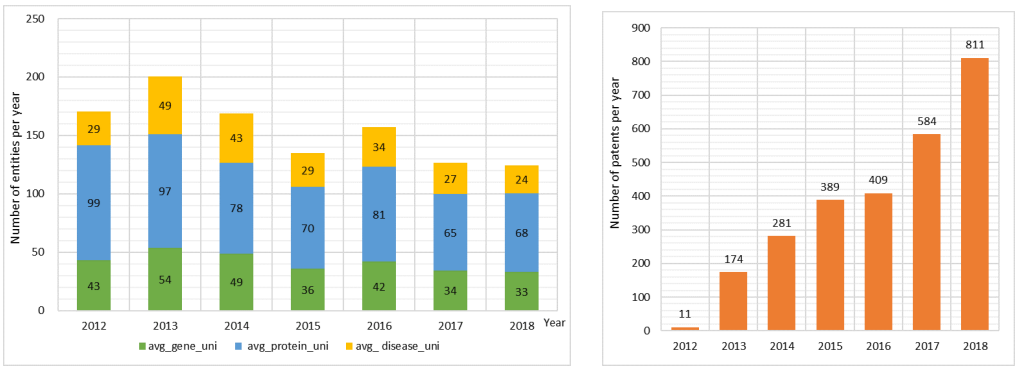
We can notice that, in both dataset, although we got most predictions of protein entities, the top 10 common protein mentions are actually least meaningful or reasonable compared with other entity types. It is interesting that with these limited labelled data and not very high NER classification performance, and just after some very simple post cleaning steps, the detected gene and disease entities are really meaningful and the most common ones also seems very reasonable. Besides, we can notice that our model can sucessfully recognize code-mixing entities, like the c-MAF基因(c-MAF gene) in top 10 common gene mentions shown in Table 4.5. This proved that our model and solution solved the code-mixing problem of Chinese biomedical patents text then.

For predictions on dataset HG, all top 10 commonly mentioned entities are quite different from dataset BC, but still has some overlappings, which are also reasonable. Similarly, although the protein entities are most commonly recognized, they seems less meaningful or reasonable, that the antibody-based drugs dominate over drug targets and protein biomarkers, for example, 免疫球蛋白(Immunoglobulin) and 贝伐单抗(Bevacizumab), which are general definition of one type of proteins or antibodies, and were also among top 10 commonly mentioned protein entities of our 2 dataset. We think this may possibly caused by that we annotated some more general protein concepts and those protein drugs in our gold standard dataset as protein entities as well.

This indicates that the style and quality of the gold standard dataset can influence the final NER classification a lot even the size of dataset is limited. One vital challenge for both annotators when building our gold standard dataset is that, distinguishing among drugs, (diagnostic) biomarkers and drug targets proteins and in which situation to annotate them as protein entity. Let alone for the language model and the machine. We can imagine that if larger and higher quality gold standard dataset is built in the future, by more annotators with professional biomedical background knowledge, this model or solution may work way better than current attempt.

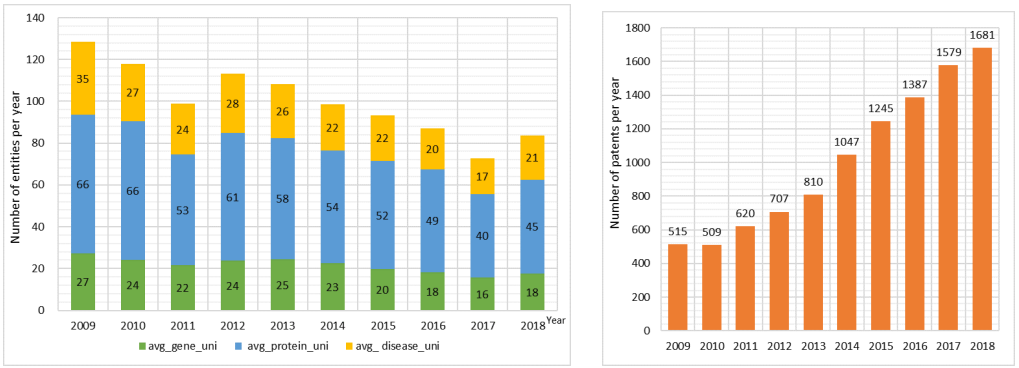
4.3.2 Year trends and technology influence

Then, the detailed entity and document information changing with years are analyzed and shown in Figure 4.1 for dataset BC and Figure 4.2 for dataset HG. The 'avg' in legend names means it shows the average value among all documents of one year, while the 'uni' means it only consider unique mentions.



(a) (b)

Figure 4.1: Year trend of predictions on dataset BC (a) Average unique biomedical entity information. (b) Number of patents.

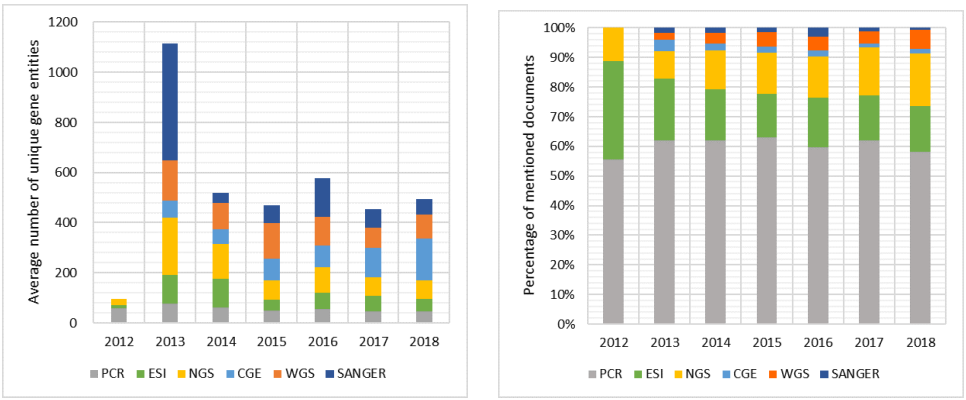


(a) (b)

Figure 4.2: Year trend of predictions on dataset HG (a) Average unique biomedical entity mentions. (b) Number of patents.

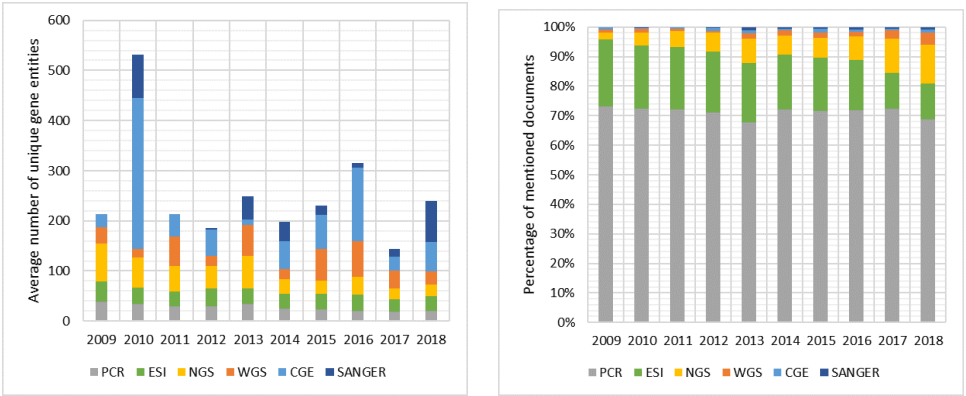
When focusing on the biomedical entity mentions situation per year of both dataset, we can notice that protein was always the most mentioned entity type every year, which was consistent with the overall entity prediction situation shown in basic analysis part. The amount of average unique mentions of each entity type did not change too much with years, and it shows a fluctuation situation as well. While on the other hand, the number of documents in each dataset keeps growing steadily every year.

We calculated the average unique gene entities detected from all patent documents mentioning one specific technology each year. We also calculated the percentage of documents which mention different technology each year. The results of these two calculations are shown in Figure 4.3 for dataset BC and Figure 4.4 for dataset HG.



(a) (b)

Figure 4.3: Biological technology influence analysis of dataset BC (a) Average unique gene entity mentions using different technologies. (b) Percentage of patents mentioning each technology.



(a) (b)

Figure 4.4: Biological technology influence analysis of dataset HG (a) Average unique gene entity mentions using different technologies. (b) Percentage of patents mentioning each technology.

The average unique gene mentions results of different technology used per year did not show obvious changing trends, which are also consistent to the biomedical entity mentions situation per year. However, this may also because we only retrieved patents no earlier than 2009, while both NGS and WGS was developed in the 1990s. If some future work can be done to include more and earlier patents, there may possibly be some different discoveries then. We can notice that, the percentage of patents graph of both dataset show a trend that, the mentions or usage of relatively new technology (NGS and WGS) keeps growing with years, while for older ones (CGE and SANGER) it slowly decreases then. And it is really obvious that PCR is always the most commonly mentioned or used technology among our selected ones all along.

4.3.3 Connection network

We calculated and generated gene-gene connection (co-occurrence) network from gene entity predictions of dataset BC (set edge weight threshold as 5) and HG dataset (set edge weight threshold as 2), respectively. The whole visualization of the two networks are given in Appendices. Here we will show part of both network and give some explanations and analysis. A part of HG dataset whole network was shown in Figure 4.5.

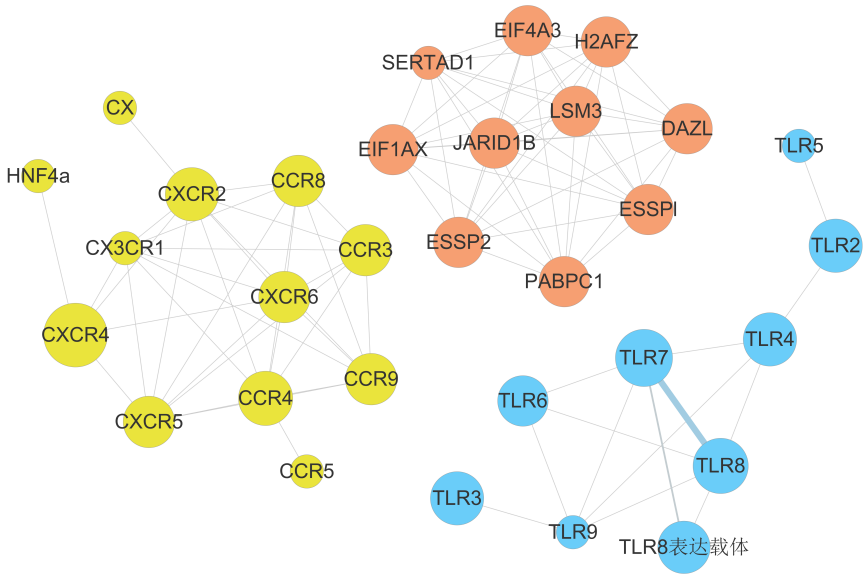


Figure 4.5: Part of gene connection clusters generated by HG dataset (edge weight threshold: 2)

We can notice there are 3 different gene clusters, which are meaningful and reasonable that all genes in the same cluster indeed have real biological interactions or close relations(e.g. from same gene family). There are also a lot other similar clusters in both whole version networks, which indicates that our predictions

and co-occurrence network mining method can indeed find relatively reliable and meaningful connections among genes.

However, from one part of the gene network generated by BC dataset, shown in Figure 4.6, we can still find some problems are unsolved during this study. One vital one is the repetitive appearance of the same gene entity node caused by different reasons.

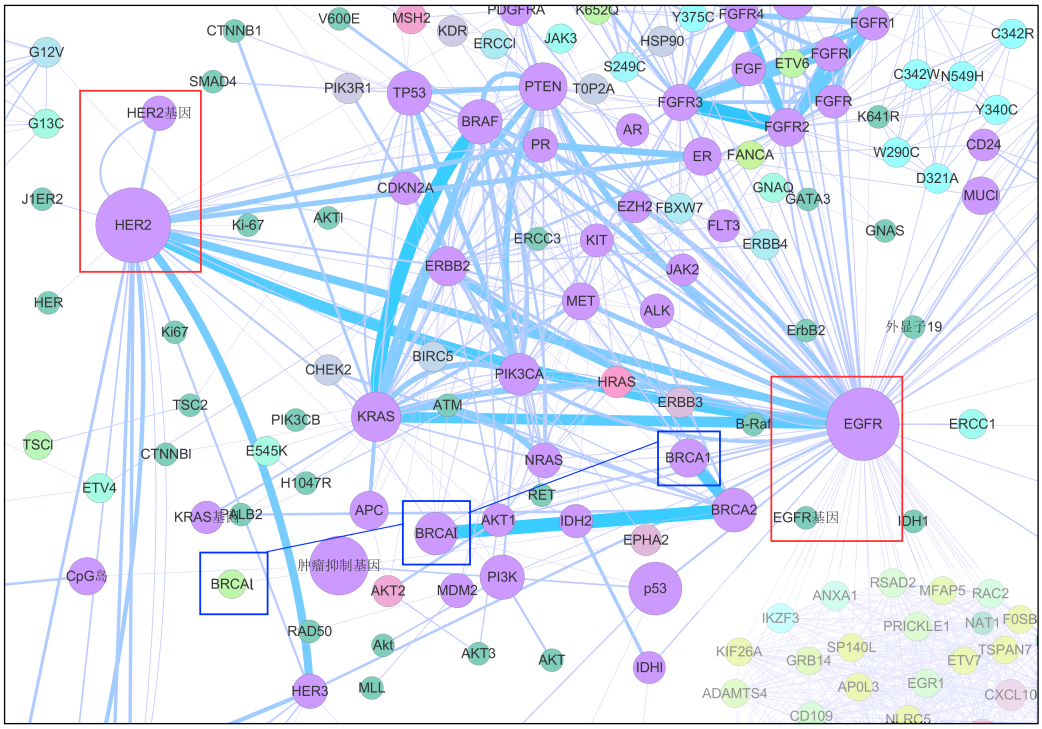
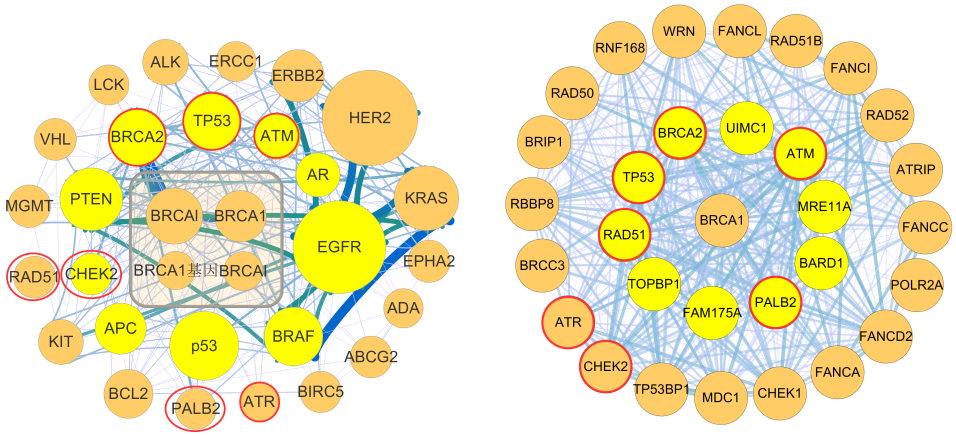


Figure 4.6: Part of gene connection network generated by BC dataset

In each of the red blocks noted on the figure, we can notice that both 2 nodes are actually referring the same gene but one of them are written in a code-mixing format with word '基因'(gene) behind the gene name, while the other without. This indicates the first reason of repetitive gene node appearance, which is the code-mixing written style of Chinese biomedical patents text. Another reason is the OCR error since our text source (Google patent HTML plain text contents) were probably derived from corresponding official Chinese patent PDF files. This can be indicated by the 3 nodes inside the dark blue blocks noted on the figure. The 3 nodes are actually referring to the same gene 'BRCA1', but the '1' in the 3 node names are written as '1', 'l' and 'I'(lowercase 'L' and uppercase 'i') respectively, leading to 3 different gene nodes then.

Besides 2 large gene connection networks, we also calculated a small cluster network only focusing on interactions among gene 'BRCA1' and its all first-degree neighbours, in order to do an example comparison using STRING database. We calculated the 'BRCA1' gene cluster from BC dataset since 'BRCA1' is a breast cancer related biomarker and BC dataset is also a breast cancer related patents dataset. The visualization of our predicted BRCA1 network (edge weight threshold: 3), plus a BRCA1 network derived from STRING database [27] with text mining edge sources (top 30 common first degree neighbours), are shown in Figure 4.7.



(a) (b)

Figure 4.7: Comparison of BRCA1 gene network (a) BRCA1 gene network generated by predictions of our BC dataset, yellow colored nodes are connected with BRCA1 with top 10 edge weights; (b) BRCA1 gene network generated by STRING databases, yellow colored nodes are top 10 common neighbours of BRCA1 in STRING. Nodes enclosed in red circle are matching ones of two networks.

As shown in the figure, the nodes connected with BRCA1 with top 10 edge weights in our predicted network match 4 genes(nodes) of top 10 common neighbours of BRCA1 in the STRING network, and 6 nodes in our predicted network match what appear in STRING BRCA1 network in total. Obviously that our predictions contains much more novel gene nodes and edges compared with STRING one. Since we already only retrieved the edges with text mining sources from STRING database, here indicates that the novel nodes and edges appeared in our predicted network are possible to be new discoveries or patented in China but not in Europe (since STRING is part of ELIXIR Core Data Resources, which are a set of European data resources[40].) However, better performed NER classifiers and more network comparisons are still needed to prove these assumptions then.

5 Conclusions and Future Work

5.1 Conclusions

In this study, we found a possible solution to solve the biomedical NER problem on the very specific domain, Chinese biomedical patents data, which is English-Chinese code-mixing and has complex text writing style. We built our own Chinese Biomedical patents dataset, including one humanly labeled gold standard dataset which contains 5,813 sentences and 2,267 unique named entities from 21 patent documents, and two large unlabeled dataset which contain 2,659 and 53,007 patent documents respectively.

We tried an example benchmark experiment to compare the LatticeLSTM and BERT pre-trained language model trained on general domain Chinese text data to solve NER tasks. We found that the LatticeLSTM did not work better than both BERT models we implemented, and need way longer training time then BERT as well. Thus we continued with only the BERT models during our training experiments. After we implemented 3 different BERT models and learning methods, we trained and evaluated them on our evaluation sets. The results showed that the BERT LM mixed model, which was trained on partHG dataset(unlabled 10,000 patents in train set, 100 in test set) for only 1 epoch, got the optimal results in all entity categories, while actually all models or training methods did not differ a lot. The best model got a 0.54 ± 0.15 micro average F1 score among all entity types (among all evaluation sets).

We finally generated some predictions with the trained best model (trained on the whole gold standard dataset), then did some further biomedical related analysis. These analysis indicates that our built solution and trained model is available to detect meaningful biomedical entities and can found some novel gene and gene-gene interactions, just with limited labeled data, training time and computing power.

5.2 Limitations and possible improvements

However, as a first attempt to solve this extremely domain-specific NER task, we did not get a very high classification performance, and still a lot possible work can be done for further improvement. As mentioned in section 3.1.3 Silver standard dataset, we finally failed to built a good silver standard dataset, which we have concluded that the mainly reason are: 1). lack of high quality and open source Chinese biomedical entity data; 2). OCR errors in original text source; 3). code-mixing written style. Based on these 3 reasons, several future work can be inspired then.

First is that, once there are more good and available Chinese biomedical entity data, especially gene and protein entity Chinese names data, it will surely increase the possibility to build high-quality silver standard dataset, which can ensure further remote-supervised learning and better usage of unlabeled data. Secondly, just as described in subsection 2.1, OCR errors problem can possibly be solved or alleviated by building a specific Chinese biomedical OCR or OCR correction tool, based on training a Language model or existed OCR frame with some domain-specific data as what was attempted in [6] and [8].

Code-mixing is actually a very common language usage style in a lot countries and areas, for example India and Hong Kong. Thus, there has been some attempts trying to solve Chinese-English code-mixing problems in general domain, and more commonly with speech data since this code-mixing usage seems more possible to happen in non-official situations, for example casual conversations or social media posts[41][42]. Thus, it will be interesting to try some code-mixing language model in general domain to check whether it can improve current NER performance then.

5.3 Extension work

Besides the above possible steps to solve current limitations and problems, there is also some extension work which may improve current solution. As we found in either our training experiment results or further analysis, the size and quality of gold-standard dataset, plus the training time/steps, can really influence the NER classifier performance a lot. Thus, although we offered a solution to solve Chinese biomedical NER task with limited labeled data, if larger and higher quality labeled dataset are available, and if more training steps on unlabeled can be ensured, it is highly possible that the model can work way better then.

In the final part of our study, we did some biological related post analysis in order to mine some meaningful information from our generated predictions. There are still a lot interesting topics of mining meaningful biological information from text mining resources, which we can try in the future. The first point is that, we built our gene-gene connection network using the co-occurrence concept, it is

interesting if further relation extraction work can be done and make comparison between connections derived by co-occurrence then.

Secondly, as mentioned in section 4.3.2 Year trends and technology influence, more patents and patents earlier than year 2000 can be involved in future attempts, in order to discover possible changes of gene and protein mentioned influenced by NGS and WGS, which were both developed in the 1990s. And more technologies can also be involved analyzing the influence, or directly try to detect biomedical technology entities during the NER process.

Besides, our analysis can then be considered as resources to build a gene or protein search platform, which can offer corresponding patent and some entity related information when search for one specific gene, protein or disease name. For example, [43] is such an AI-Assisted antibody searching platform which did the similar thing. These type of platforms can then save a lot less useful or less novel topics when start a project and will definitely inspire some interesting new studies as well.

Bibliography

[1] Maryam Habibi, Leon Weber, Mariana Neves, David Luis Wiegandt, and Ulf Leser. Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, 33(14):i37–i48, 2017.

[2] J-D Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl_1):i180–i182, 2003.

[3] Ming-Siang Huang, Po-Ting Lai, Richard Tzong-Han Tsai, and Wen-Lian Hsu. Revised jnlpba corpus: A revised version of biomedical ner corpus for relation extraction task. *arXiv preprint arXiv:1901.10219*, 2019.

[4] Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. Introduction to the bio-entity recognition task at jnlpba. In *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications*, pages 70–75. Citeseer, 2004.

[5] Li Du. Patenting human genes: Chinese academic articles' portrayal of gene patents. *BMC medical ethics*, 19(1):29, 2018.

[6] Eva D'hondt, Cyril Grouin, and Brigitte Grau. Low-resource ocr error detection and correction in french clinical texts. In *Proceedings of the Seventh International Workshop on Health Text Mining and Information Analysis*, pages 61–68, 2016.

[7] Paul Thompson, John McNaught, and Sophia Ananiadou. Customised ocr correction for historical medical text. In *2015 Digital Heritage*, volume 1, pages 35–42. IEEE, 2015.

[8] Congyue ZHANG, Ziming YIN, Dayun SUN, and Wei DAI. Recognition technology of the laboratory sheet based on tesseract. *Beijing Biomedical Engineering*, (3):11, 2019.

[9] Hye-Jeong Song, Byeong-Cheol Jo, Chan-Young Park, Jong-Dae Kim, and Yu-Seop Kim. Comparison of named entity recognition methodologies in biomedical documents. *Biomedical engineering online*, 17(2):158, 2018.

[10] Golnar Sheikhshab, Inanc Birol, and Anoop Sarkar. In-domain context-aware token embeddings improve biomedical named entity recognition. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, pages 160–164, 2018.

[11] Thanh Hai Dang, Hoang-Quynh Le, Trang M Nguyen, and Sinh T Vu. D3ner: biomedical named entity recognition using crf-bilstm improved with fine-tuned embeddings of various linguistic information. *Bioinformatics*, 34(20):3539–3546, 2018.

[12] Wonjin Yoon, Chan Ho So, Jinhyuk Lee, and Jaewoo Kang. Collabonet: collaboration of deep neural networks for biomedical named entity recognition. *BMC bioinformatics*, 20(10):249, 2019.

[13] Qianhui Lu, Yunlai Xu, Runqi Yang, Ning Li, and Chongjun Wang. Serial and parallel recurrent convolutional neural networks for biomedical named entity recognition. In *International Conference on Database Systems for Advanced Applications*, pages 439–443. Springer, 2019.

[14] Larry Smith, Lorraine K Tanabe, Rie Johnson nee Ando, Cheng-Ju Kuo, I-Fang Chung, Chun-Nan Hsu, Yu-Shi Lin, Roman Klinger, Christoph M Friedrich, Kuzman Ganchev, et al. Overview of biocreative ii gene mention recognition. *Genome biology*, 9(2):S2, 2008.

[15] Jiao Li Xuwen Wang and Junlian Li Yingjie Wu. Bilstm-crf based open concept relation extraction from chinese biomedical texts. *Chinese Journal of Medical Library and Information Scinence*, 27(11):33–39, 2019.

[16] Yue Zhang and Jie Yang. Chinese ner using lattice lstm. 2018.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[18] ProHiryu. Use the pre-trained bert language model to do chinese ner. `https://github.com/ProHiryu/bert-chinese-ner`.

[19] Gina-Anne Levow. The third international chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 108–117, 2006.

[20] Kui Xue, Yangming Zhou, Zhiyuan Ma, Tong Ruan, Huanhuan Zhang, and Ping He. Fine-tuning bert for joint entity and relation extraction in chinese medical text. *arXiv preprint arXiv:1908.07721*, 2019.

[21] Fei Song and W Bruce Croft. A general language model for information retrieval. In *Proceedings of the eighth international conference on Information and knowledge management*, pages 316–321. ACM, 1999.

[22] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*, 2010.

[23] Erik F Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*, 2003.

[24] HUGO Gene Nomenclature Committee. Hgnc custom gene names search and download engine. https://www.genenames.org/download/custom/.

[25] The Universal Protein Resource. Uniprot protein sequence and annotation database. https://www.uniprot.org/.

[26] World Health Organization. Icd-11 homepage. https://www.who.int/classifications/icd/en/.

[27] String. Protein-protein interaction networks functional enrichment analysis. https://string-db.org/.

[28] Google. Google patents: Search and read the full text of patents from around the world. https://patents.google.com/.

[29] gov.cn. Cnipa: Patents search and analysis. http://www.pss-system.gov.cn/.

[30] World Intellectual Property Organization. International patent classification (IPC). https://www.wipo.int/classifications/ipc/en/.

[31] chakki. doccano: Open source text annotation tool for machine learning practitioner. https://github.com/chakki-works/doccano.

[32] Jie Yang. Yedda: A lightweight collaborative text span annotation tool. https://github.com/jiesutd/YEDDA.

[33] National Health Commission of the People's Republic of China. ICD-11 Chinese translated version. http://www.nhc.gov.cn/ewebeditor/uploadfile/2018/12/20181221160228191.xlsx.

[34] Alfred V Aho and Margaret J Corasick. Efficient string matching: an aid to bibliographic search. *Communications of the ACM*, 18(6):333–340, 1975.

[35] Zalando Research. Flair: A very simple framework for state-of-the-art nlp. https://github.com/flairNLP/flair.

[36] Cyril Grouin, Sophie Rosset, Pierre Zweigenbaum, Karën Fort, Olivier Galibert, and Ludovic Quintard. Proposal for an extension of traditional named entities: From guidelines to evaluation, an overview. In *Proceedings of the 5th linguistic annotation workshop*, pages 92–100. Association for Computational Linguistics, 2011.

[37] Jie Yang. Lattice lstm original project repository. https://github.com/jiesutd/LatticeLSTM.

[38] Kyubyong Park. Pytorch implementation of ner with pretrained bert. https://github.com/Kyubyong/bert_ner.

[39] Hugging Face. Pytorch tansformers by hugging face. https://github.com/huggingface/transformers.

[40] ELIXIR Core Data Resources. A set of european data resources of fundamental importance to the wider life-science community and the long-term preservation of biological data. https://elixir-europe.org/platforms/data/core-data-resources.

[41] Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. Code-switching language modeling using syntax-aware multi-task learning. *arXiv preprint arXiv:1805.12070*, 2018.

[42] Han-Ping Shen, Chung-Hsien Wu, Yan-Ting Yang, and Chun-Shan Hsu. Cecos: A chinese-english code-switching speech database. In *2011 International Conference on Speech Database and Assessments (Oriental COCOSDA)*, pages 120–123. IEEE, 2011.

[43] BenchSci. Ai-assisted antibody selection. https://www.benchsci.com/.

A Appendices

A.1 Detailed IPC code filters

The filters were set based on analysis on our patents data. We would first determine whether a patent contain codes from the 'exception codes list', if does, the patent would be discarded; if not, we then need to determine whether the patent contains codes from the 'kept codes list', if does, keep it, else discard.

kept codes for dataset BC: A61, C07, C12N, C12Q, G01N, G16B, G16C, G16H kept codes for dataset HG: A61, C07, C12N, C12Q, C12Y, C12P, C12M, G01N, G16B, G16C, G16H Exception codes for both dataset: A01, A21, A23, B09, C02, C09, C10, C11, C05, H01, H04, Y02

A61: medical or veterinary science; hygiene;
C07: organic chemistry;
C12N: microorganisms or enzymes; compositions thereof; propagating, preserving or maintaining microorganisms; mutation or genetic engineering; culture media;
C12Q: measuring or testing processes involving enzymes, nucleic acids or microorganisms; compositions or test papers therefor; processes of preparing such compositions; condition-responsive control in microbiological or enzymological processes;
C12Y: enzymes;
C12P: fermentation or enzyme-using processes to synthesise a desired chemical compound or composition or to separate optical isomers from a racemic mixture;
C12M: apparatus for enzymology or microbiology; apparatus for culturing microorganisms for producing biomass, for growing cells or for obtaining fermentation or metabolic products, i.e. bioreactors or fermenters;
G01N: investigating or analysing materials by determining their chemical or physical properties;
G16B: bioinformatics, i.e. information and communication technology [ict] specially adapted for genetic or protein-related data processing in computational molecular biology;
G16C: computational chemistry; chemoinformatics; computational materials science;
G16H: healthcare informatics, i.e. information and communication technology [ict] specially adapted for the handling or processing of medical or healthcare data;

Y02: technologies or applications for mitigation or adaptation against climate change;
A01: agriculture; forestry; animal husbandry; hunting; trapping; fishing;
A23: foods or foodstuffs; their treatment, not covered by other classes;
C02: treatment of water, waste water, sewage, or sludge;
C09: dyes; paints; polishes; natural resins; adhesives; compositions not otherwise provided for; applications of materials not otherwise provided for;
C11: animal or vegetable oils, fats, fatty substances or waxes; fatty acids therefrom; detergents; candles;
C05: fertilisers; manufacture thereof;
H01: basic electric elements;

H04: electric communication technique;
B09: disposal of solid waste; reclamation of contaminated soil;
A21: baking; edible doughs;
C10: petroleum, gas or coke industries; technical gases containing carbon monoxide; fuels; lubricants; peat;

A.2 Detailed synonym list of each biological technology

'NGS': NGS, next-generation-sequencing, 二代测序, 第二代DNA测序, 下一代测序, Illumina, illumina;

'WGS': WGS, whole-genome-sequencing, 全基因组测序;

'PCR': PCR, polymerase-chain-reaction, qPCR, 聚合酶链式反应, 多聚酶链式反应;

'ESI': ESI, electrospray-ionization, 电喷雾;

'CGE': CGE, capillary-gel-electrophoresis, 毛细管凝胶电泳;

'SANGER': 桑格, sanger-sequencing.

A.3 Detailed information of cross-validation-like evaluation datasets

Table A.1: Datasets for cross-validation-like evaluation

(train/test/dev)	**gene_total**	**gene_unique**	**protein_total**	**protein_unique**
cpb00*	1756/121/11	453/20/9	4984/3/43	1018/3/33
cpb01	1518/16/354	355/2/125	3236/158/1636	780/110/178
cpb02	1852/36/0	458/25/0	4941/89/0	1026/30/0
cpb03	1130/737/21	319/157/8	4865/157/8	974/81/2
cpb04	1829/26/33	452/10/23	4475/472/83	952/88/25
(train/test/dev)	**disease_total**	**disease_unique**	**all_total**	**all_unique**
cpb00	1756/73/8	728/2/6	9398/197/62	2199/25/48
cpb01	2583/156/0	702/53/0	7337/330/1990	1837/165/303
cpb02	2006/709/24	560/230/2	8799/834/24	2044/285/2
cpb03	2402/337/0	684/78/0	8397/1231/29	1977/316/10
cpb04	2035/177/527	511/103/194	8339/675/643	1915/201/242

* cbp stands for Chinese Biomedical Patents.

A.4 Detailed cross-validation-like experiments results

Table A.2: Detailed training experiments results

PartBERT+CRF fine-tuning																	
dataset	**cbp00**			**cbp01**			**cbp02**			**cbp03**			**cbp04**				
measure	**pre**	**rec**	**f1**	**pre**	**rec**	**f1**	**pre**	**rec**	**f1**	**pre**	**rec**	**f1**	**pre**	**rec**	**f1**	**avg_f1**	**std_f1**
gene	0.65	0.17	0.26	0.17	0.50	0.25	0.64	0.19	0.30	0.39	0.07	0.12	0.12	0.12	0.12	0.21	*0.08
protein	0.00	0.00	0.00	0.32	0.17	0.22	0.30	0.13	0.19	0.47	0.23	0.31	0.80	0.42	0.55	0.25	0.20
disease	0.79	0.96	0.86	0.48	0.65	0.55	0.89	0.53	0.66	0.89	0.56	0.69	0.67	0.44	0.61	0.67	0.12
macro avg	0.48	0.38	0.37	0.32	0.44	0.34	0.61	0.28	0.38	0.58	0.29	0.37	0.53	0.33	0.43	0.38	*0.03
micro avg	0.75	0.46	0.46	0.40	0.41	0.40	0.83	0.47	0.60	0.66	0.22	0.33	0.67	0.47	0.55	0.47	*0.11
Supervised Original																	
dataset	**cbp00**			**cbp01**			**cbp02**			**cbp03**			**cbp04**				
measure	**pre**	**rec**	**f1**	**pre**	**rec**	**f1**	**pre**	**rec**	**f1**	**pre**	**rec**	**f1**	**pre**	**rec**	**f1**	**avg_f1**	**std_f1**
gene	0.62	0.52	0.57	0.35	0.44	0.39	0.35	0.47	0.40	0.14	0.09	0.11	0.05	0.19	0.08	*0.31	0.21
protein	0.16	0.86	0.27	0.44	0.41	0.42	0.24	0.45	0.31	0.19	0.26	0.22	0.56	0.46	0.50	0.34	*0.11
disease	0.60	0.82	0.69	0.52	0.64	0.57	0.78	0.65	0.71	0.68	0.46	0.55	0.44	0.55	0.49	0.60	*0.09
macro avg	0.46	0.73	0.51	0.44	0.50	0.46	0.46	0.52	0.47	0.34	0.27	0.29	0.35	0.40	0.36	0.42	0.09
micro avg	0.60	0.64	0.61	0.47	0.52	0.49	0.70	0.62	0.65	0.29	0.21	0.24	0.51	0.47	0.48	0.49	0.16
LM mixed fine-tuning (smallBC_1epoch)																	
dataset	**cbp00**			**cbp01**			**cbp02**			**cbp03**			**cbp04**				
measure	**pre**	**rec**	**f1**	**pre**	**rec**	**f1**	**pre**	**rec**	**f1**	**pre**	**rec**	**f1**	**pre**	**rec**	**f1**	**avg_f1**	std_f1
gene	0.40	0.24	0.30	0.00	0.00	0.00	0.56	0.42	0.48	0.20	0.11	0.14	0.09	0.31	0.14	0.21	0.18
protein	0.00	0.00	0.00	0.50	0.53	0.52	0.21	0.35	0.26	0.26	0.40	0.31	0.54	0.57	0.56	0.33	0.23
disease	0.84	0.99	0.91	0.52	0.65	0.58	0.80	0.72	0.76	0.71	0.48	0.58	0.44	0.64	0.52	0.67	0.16
macro avg	0.41	0.41	0.40	0.34	0.39	0.37	0.52	0.50	0.50	0.39	0.33	0.34	0.36	0.51	0.41	0.40	0.06
micro avg	0.55	0.51	0.52	0.49	0.56	0.52	0.73	0.67	0.70	0.35	0.25	0.28	0.50	0.58	0.53	0.51	0.15
LM mixed fine-tuning (smallBC_30epochs)																	
dataset	**cbp00**			**cbp01**			**cbp02**			**cbp03**			**cbp04**				
measure	**pre**	**rec**	**f1**	**pre**	**rec**	**f1**	**pre**	**rec**	**f1**	**pre**	**rec**	**f1**	**pre**	**rec**	**f1**	**avg_f1**	**std_f1**
gene	0.35	0.31	0.33	0.29	0.31	0.30	0.53	0.53	0.53	0.12	0.10	0.11	0.04	0.08	0.05	0.26	0.19
protein	0.01	0.33	0.02	0.51	0.57	0.54	0.22	0.31	0.26	0.38	0.32	0.35	0.60	0.65	0.63	*0.36	0.24
disease	0.79	0.90	0.84	0.51	0.62	0.56	0.83	0.69	0.75	0.77	0.52	0.62	0.49	0.68	0.57	0.67	0.12
macro avg	0.38	0.51	0.40	0.44	0.50	0.47	0.53	0.51	0.51	0.42	0.31	0.36	0.38	0.47	0.42	*0.43	0.06
micro avg	0.51	0.53	0.51	0.59	0.58	0.54	0.75	0.65	0.69	0.33	0.24	0.28	0.55	0.64	0.59	0.52	0.15
LM mixed fine-tuning (partHG_1epoch)																	
dataset	**cbp00**			**cbp01**			**cbp02**			**cbp03**			**cbp04**				
measure	**pre**	**rec**	**f1**	**pre**	**rec**	**f1**	**pre**	**rec**	**f1**	**pre**	**rec**	**f1**	**pre**	**rec**	**f1**	**avg_f1**	**std_f1**
gene	0.52	0.40	0.46	0.15	0.31	0.20	0.44	0.61	0.51	0.18	0.11	0.14	0.02	0.04	0.02	0.27	0.21
protein	0.00	0.00	0.00	0.38	0.53	0.44	0.24	0.38	0.30	0.32	0.32	0.32	0.61	0.59	0.60	0.33	0.22
disease	0.78	0.92	0.84	0.58	0.89	0.71	0.80	0.73	0.76	0.72	0.57	0.64	0.46	0.64	0.53	*0.70	0.12
macro avg	0.43	0.44	0.43	0.37	0.58	0.45	0.49	0.57	0.52	0.41	0.33	0.37	0.36	0.42	0.38	*0.43	0.06
micro avg	0.61	0.59	0.59	0.46	0.69	0.55	0.72	0.69	0.70	0.35	0.26	0.30	0.55	0.58	0.56	*0.54	0.15

A.5 Whole version of gene co-occurrence networks

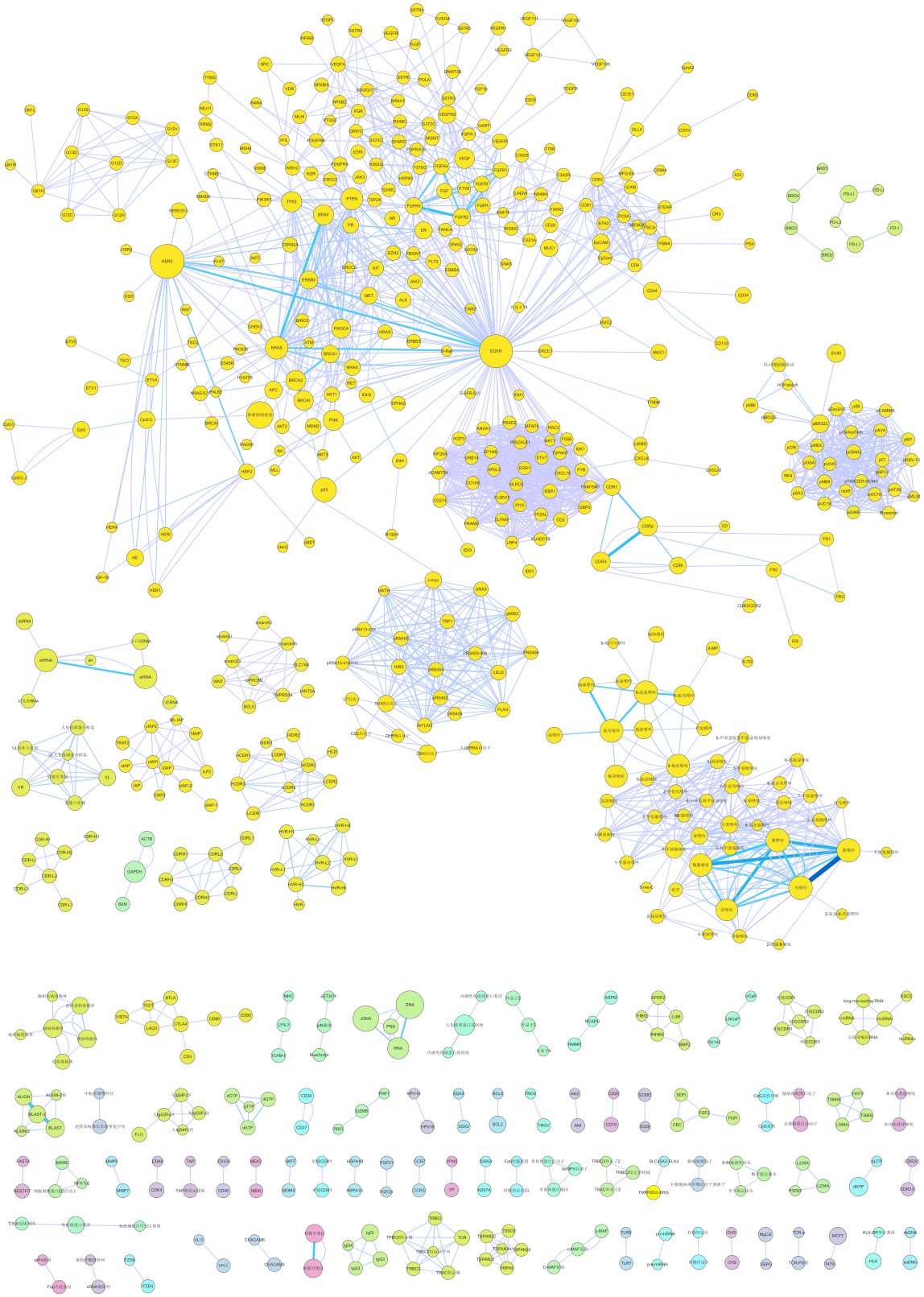


Figure A.1: Gene connection network generated by BC dataset (edge weight threshold: 5)

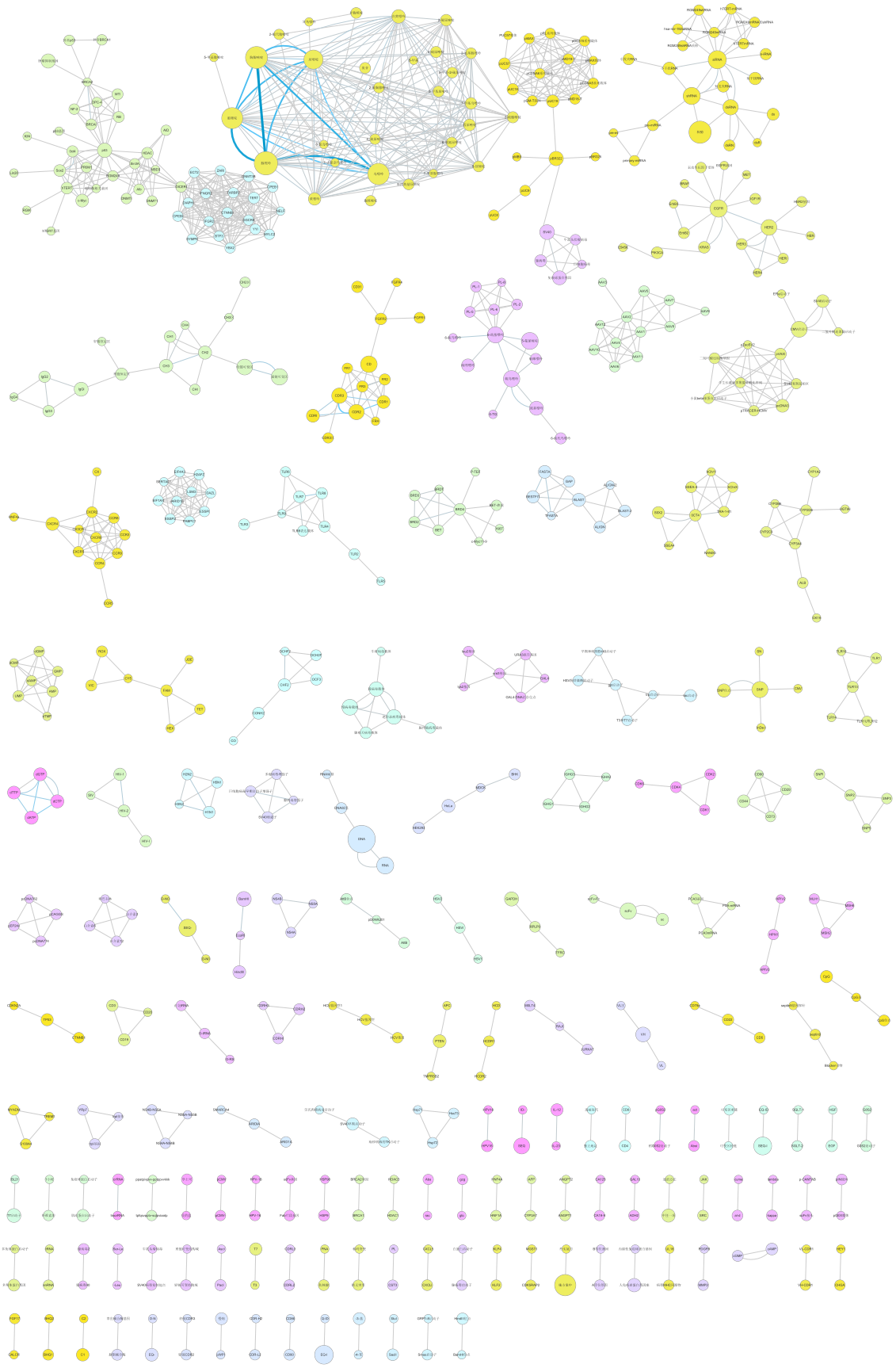


Figure A.2: Gene connection network generated by HG dataset (edge weight threshold: 2)