Spring 5-19-2020

Improved Chinese Language Processing for an Open Source Search Engine

Xianghong Sun
San Jose

Improved Chinese Language Processing for an Open Source Search Engine

A Project

Presented to

The Faculty of the Department of Computer Science

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

By

Xianghong Sun

May 2020

The Designated Committee Approves the Project Titled

Improved Chinese Language Processing for an Open Source Search Engine

by

Xianghong Sun

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSÉ STATE UNIVERSITY

May 2020

Dr. Chris Pollett Department of Computer Science

Dr. Robert Chun Department of Computer Science

Dr. Mike Wu Department of Computer Science

Acknowledgement

First, I would like to thank Dr. Chris Pollett, my advisor, for the help, guidance through the project.

Second, I would like to thank my committee members, Dr. Robert Chun, and Dr. Mike Wu, for all the time and feedbacks for my project.

Last, I would like to thank my family for all the support for my academics.

ABSTRACT

Natural Language Processing (NLP) is the process of computers analyzing on human languages. There are also many areas in NLP. Some of the areas include speech recognition, natural language understanding, and natural language generation.

Information retrieval and natural language processing for Asians languages has its own unique set of challenges not present for Indo-European languages. Some of these are text segmentation, named entity recognition in unsegmented text, and part of speech tagging. In this report, we describe our implementation of and experiments with improving the Chinese language processing sub-component of an open source search engine, Yioop. In particular, we rewrote and improved the following sub-systems of Yioop to try to make them as state-of-the-art as possible: Chinese text segmentation, Part-of-speech (POS) tagging, Named Entity Recognition (NER), and Question and Answering System.

Compared to the previous system we had a 9% improvement on Chinese words Segmentation accuracy. We built POS tagging with 89% accuracy. And We implement NER System with 76% accuracy.

Key words: Natural Language Processing, Chinese, Chinese Words Segmentation, Part-of-Speech Tagging, Named Entity Recognition, Question Answering System

TABLE OF CONTENTS

List of Figures

Introduction

Almost everyone uses search engines, such as Google and Bing. When users search something in a search engine, the search engine needs to fetch the related documents by looking for the keywords and return the results in relatively short time. Yioop is an open source search engine software written in PHP. It provides many features, such as search results, media services, social groups, blogs, and wikis. In order to process the documents from different languages, many NLP techniques are needed. For example, if the user is entering a question in Chinese, the engine needs to segment the sentence, generate the key words, fetch the database to find the results, and extra the answer, etc. How good the engine can process depends on how good each feature is implemented.

In this project, four features are implemented. They are focused in Chinese language, but they can be extended to other languages. Four features include term Segmentation, POS tagging, NER, and Question and Answering. In this paper, I am going to introduce how those four features are implemented, what models do I use, and how accurate they are one after another.

It is natural that the word segmentation is the first step in Chinese NLP. Different from most western language, some Asian languages, such as Chinese or Japanese, do not have a word delimiter between words. English has spaces between words while Chinese does not. Chinese text consists of a continues string of Chinese characters, which causes ambiguous. As a native Chinese, I sometimes feel hard to segment some strings of Chinese text. Since it is the beginning step in the process, it is overall the

most important feature. If it does not work properly, such as having a low accuracy, the other features that depends on it will also have a low accuracy.

The algorithm for Chinese Segmentation in current Yioop system is reverse maximum matching algorithm. It is one of the most efficient algorithms to segment Chinese, but it does not have a good accuracy compared to the latest research. So, I improved this feature by implementing a Stochastic Finite-State Word-Segmenter that segments Chinese based on the term weights. It significantly improves the current Chinese Segmentation System.

The Chinese Part-of-Speech tagging is also an important feature in Chinese NLP. It does not directly solve any NLP problem, but it is very helpful in approaching other tasks. And, it would be used in my Question and Answering System to rank the keywords of the questions. Some words are more important than other words in questions. For example, nouns are the most important words, while adjectives and verbs are less important.

There was no POS tagger for Chinese in Yioop System. The POS tagger for English are rule-based (ripple-down rules). Ripple-down rules have a very complex structure. It can figure out the POS tag by searching the surrounding words. Currently, most of the researches of POS tagging focuses on Machine Learning (ML). However, Yioop System did not have the right ML libraries for me to do this way. So, I implemented a simple machine learning model and train the data.

The Named Entity Recognizer (NER) is a feature that categories the entities in the text, mainly including person names, locations, and organization names. In more complex

system, there are more types to classify. The usage of the NER is to let computer “understand” what the text is about. Is the text telling about a person, a location, an organization, or some of them? In question and answering system, it is very important to figure out what the question is asking for.

Yioop system did not have a well-behaved named entity recognition system. It crawled Wikipedia and indexed all the titles to be the entity names. However, it had no idea whether the entity was a name, location or organization. In machine learning, NER is considered a classifier, so I built another simple ML model to approach this issue.

The last and most difficult feature in this project is the question and answering system. It combines all the features from above to make it work. A Question and Answering System is to automatically answer questions asked by human in natural languages. The basic steps of this system are defining the question type, searching related documents, and extracting the answer. My project focused on the factoid questions, so it will not answer very complex questions.

Yioop system did have a basic question and answering system, called triplet. It is a knowledge-based question answering system. The knowledges are retrieved from webs. It figures out the subject, predicate, object of the sentence. Some of the questions can be answered by this format. But this system exists a problem that it does not know what the focus of question is. For example, “Who is John Smith?” might be answered as “John Smith is 27 years old” because the answer follows the syntax of “John Smith is …”.

The new QA system improves the focus or relation of the question. It has the entity classifier as well as the relation classifier.

This report consists in five different chapters. First part states the problem, table of content and contribution of this project. Second part discuses the background of each sub-areas and its advantages and disadvantages. Third chapter deals with the designing, modeling, and implementing of the project in detail. Fourth chapter talks about the experimental results. Final chapter summarize the project and future work.

Background and Previous Work

In order to understand what happens in my project. Some background and previous works are needed for the decision making. Let us review some of the most widely used works systems nowadays.

1. Chinese Word Segmentation

As what was stated about previously, the segmentation is the first and most important step in Chinese language processing. How good the segmenter can segment the Chinese text can directly influence the success of the understanding of the text. Many methods have been tried to address the problem. Depending on the approaches, they mainly fall into four categories, which are purely statistical approaches, purely dictionary-based approaches, statistical and dictionary-based approaches, and machine learning approaches.

In purely dictionary-based approaches, mutual information and association measures of the adjacent characters are used. If the adjacent characters have higher measure than a threshold, then they are within a word because these two characters are strongly associated. Otherwise, if it does not pass the threshold, then it is the boundary of the word. In this approach, there is no need to obtain a segmented text to train. That ss an advantage over other methods. The performance of this approach is overall the poorest approaches, among others.

Purely dictionary-based approach is one of the most popular approaches. They are widely used. The advantage of it is that they are very fast and have good accuracy. The most successful algorithms are family of the maximum matching algorithms. A

maximum matching algorithm matches the longest string it can find, so it is a greedy algorithm. In this maximum matching algorithm family, forward maximum matching and backward maximum matching are two of them. Just as described in their name, the first one starts to match from the beginning of a sentence (from left to right), and the second one starts to match from the end of a sentence (from right to left). For example, there is a sentence $c_1c_2…c_n$. The forward one will start to match c_1, if c_1 is in the dictionary or it is the beginning part of a word in the dictionary, then we keep matching. We stop when the word is no longer matching anything in the dictionary, then we output the last word found in the dictionary. There is still ambiguous in this approach. And, in practice, the backward maximum matching does a better job in this family. In addition, the out-of-vocabulary words cannot be segmented correctly due to the limit of this approach. So, the completeness of the dictionary, especially person names, organization names, and location, are directly associated with the success of this approach.

Statistical dictionary-based approaches combine the previous two approaches. In short, it calculates the maximum likelihood of the combination of words appear in the sentence. For example, there are two ways to segment “日文章鱼”: “日文 章鱼” (“octopus in Japanese”). “日 文章 鱼”(“day article fish”). Assume we have the frequency of the words already; we find product of the frequency of those words and find the highest one. So, we compare Frequency(日)*Frequency(文章)*Frequency(鱼) and Frequency(日文)*Frequency(章鱼), and usually the later one wins. In practice, we sum the negative log of the frequency to find the minimum likelihood to avoid losing precision

if the numbers are getting too low. This approach is more accurate than the previous approaches, but it still cannot find the out-of-vocabulary words.

Machines Learning (ML) approaches are more popular nowadays. There are many good ML models (from early maximum entropy model to recent conditional random field model) to segment Chinese and I cannot state every of them here. So, I will generally discuss how they work here. Regardless of the model different papers use, goals are very similar. In the simplest model, it classifies whether the current character is the beginning of the word or not. If it is the beginning of the word, then tag it with B(begin); otherwise, tag it with E(end). There are more complex models that have more tags, such as B(begin), M(middle), E(end). But overall, they are very similar. The features it uses includes the combination of characters around it, some previous tags, whether it is a punctuation, and whether it is a number, etc. By far the ML approaches have the best accuracy and even can find many out-of-vocabulary words. But it might take a very long time to train and need huge amount of memory and space for the feature weight. In addition, it is much slower than previous approaches because of the calculation.

These are all different approaches in Chinese text segmentation. It is not always true that the ML approaches are the best one. Although they have higher accuracy, they are far slower.

2. Chinese Part-of-Speech tagging

From Baidu Baike, a popular Chinese search engine, there are commonly 14 Part-of-Speech tags in Chinese: noun, verb, adjective, numbers, measure word,

pronoun, adverb, preposition, conjunction, auxiliary word, interjection, mood word, onomatopoeia. But in real dataset, there are more categories and sub-categories because some special words have special categories. In Penn - CU Chinese Treebank Project, there are 41 POS tags for Chinese text. Detailed guidelines [5] can be found on their website.

There are many approaches for POS tagging in English language. The early year approaches include linguistic analysis, such as syntax, morphology, semantics. A word ends with -ly is more likely to be an adverb. However, these approaches do not do well until Machine Learning approaches come out.

Hidden Markov Model (HMM) came out in mid-1980s. Higher order HMM is good at dealing with sequence probabilities. For example, if there is a noun followed by verb, it is very likely the next term is a preposition, article or noun, but less likely to be another verb. This approach does incredibly well.

Most POS tagging approaches are static based, but we still find some rule-based approaches. RDRPOSTagger uses a ripple-down rules tree structure for the tagging process. The structure mainly consists of two parts (or branches): if-not and except. If the condition “if” holds, it will go to except and keeps checking the next condition; otherwise, it will check if-not until there is no more branch. A visual tree structure can be found in figure x. This approach unexpectedly has very high accuracy without any statistical analysis, so it is worth to discuss it here.

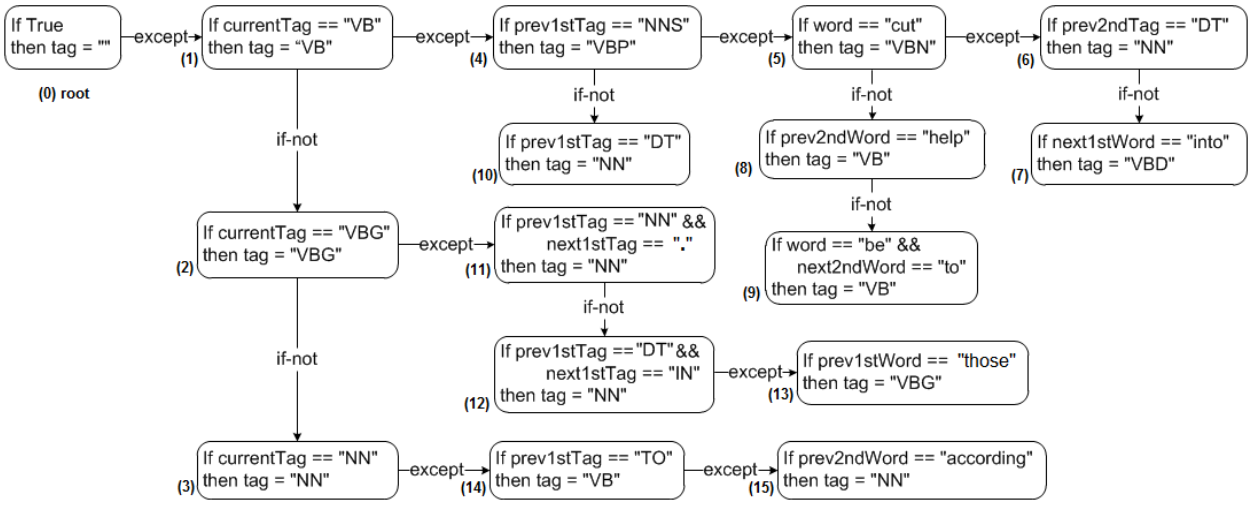

Figure 1: Ripple-Down Rule POS tagger

There is also an interesting approach that uses unsupervised ML. It uses an untagged corpus as the training data. It learns from observed patterns in the text and derive the POS tag itself. Given enough iteration, similar classes of POS words will merge. I guess this is similar to what happened in wordnet.

There are ML approaches for this task in more recent days. Support-vector machine, max entropy, and conditional random field are good examples. The features used in ML models are usually the words and tags around them.

There are more approaches than discussed here. All of them have their only advantages and disadvantages. Some of them might have fast process speed while others have higher accuracy.

3. Named Entity Recognition

Named entity recognition is a classification which identity the what categories the entity belongs to. For example, United States of American is a location. Washington can

be a name, or it can be a location depends on the context. San Jose State University is an organization. The entity might consist of more than one word. Recognizing the named entities can help computer understand the meaning of the text. The approaches for named entity recognition are usually divided into two categories: linguistic grammar-based techniques and statistical models.

Linguistic grammar-based are usually crafted by hand. It can achieve good precision. However, it might take months for many experts to makes rules. And once a new word come out, it might not be able to figure it out.

Statistical-based uses machine learning methods to train annotated texts. Maximum Entropy and Conditional Random Fields can be a reasonable choice.

4. Question and Answering System

Just like the sub-title stated, Question and Answering (QA) System is a system that automatically answer the questions by human in a natural language. There are mainly two type of QA system. First one is called knowledge-based QA, other one is IR-based QA. Most modern QA systems are hybrid systems of those two.

In first part of the QA system, whatever it is a knowledge-based or IR based, is the question processing. What is the answer type of the question? Is that a factoid question, a list question, or other type of question? The system needs to find out the entity and focus of the question in order to process the next steps.

Knowledge-based QA approaches are to build a semantic representation of query. We need a query structure to represent times, dates, locations, entities, etc. An easy representation is called semantic triple, or simply triple. For example, “Barak

Obama was born in Honolulu" can be represented by (BarakObama, PlaceOfBirth, Honolulu). We can think the triple is a structure of (subject entity, relation(focus), object entity). When answering questions, the keyword entity is search and the relation is classified to figure out which is the most related focus.

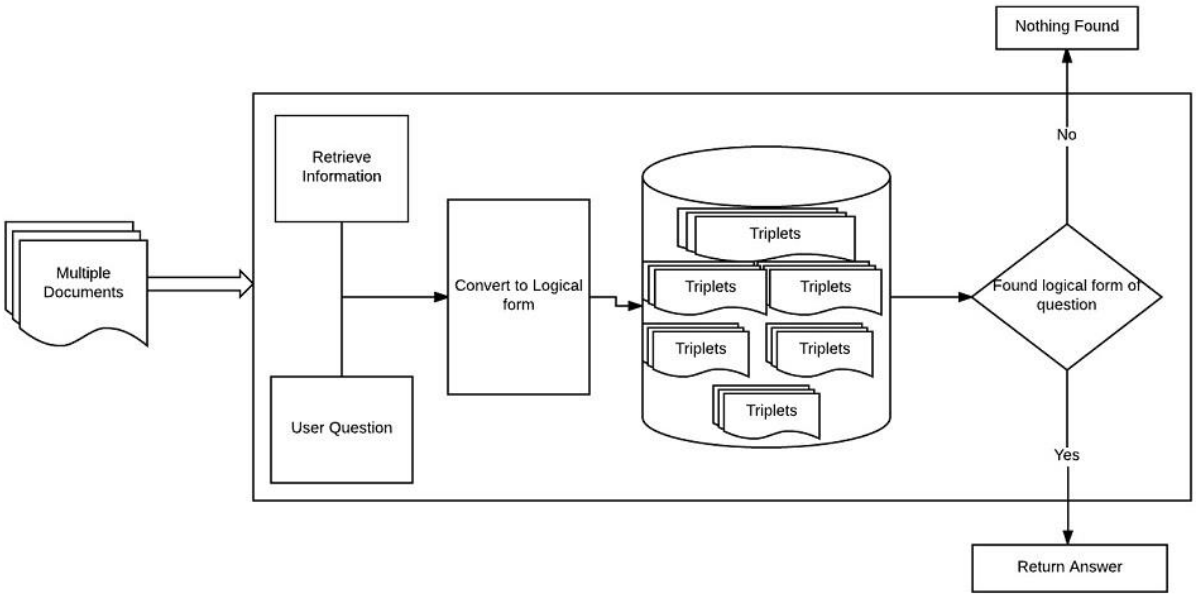

Figure 2: Knowledge-based Question Answering System

IR-based QA approaches are more complexed. It has more steps than knowledge-based approaches. The first step is the same as the knowledge-based one. Then it needs to generate query to retrieve documents. The query consists of the key words of the question. After that, it needs to find out the passage candidates. It needs to rank what passages are more likely contains the answers. Final step is to extract answer from the passages. Extracting answers from passages uses techniques of machines learning. A simple model is to train the start point and end point of the

answer. We know the answer might contained inside of a paragraph. We want to find a segment of words that best answer the question.

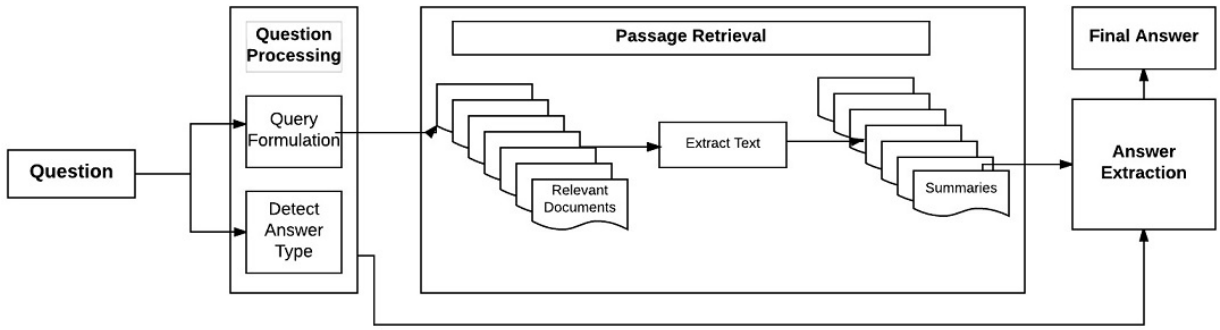

Figure 3: IR-based Question Answering System

The last approach is the hybrid approach. It tries to combine the two approaches and evaluate each other. The first half of this approach is exactly the same as the IR-based. However, in IR-based approach, we might get more than one answer from different paragraphs. The knowledge-based triple can improve the answer by verifying the which one might have a better chance to be the answer.

Design & Implementation

1. Chinese segmentation.
    i. Introduction of the model

    The model is called Stochastic Finite-state Word-segmentation. It is a statistic and dictionary-based model. There are serval reasons I considered this design:

    a. It is very fast both in training and segmenting. Different from machine learning approaches, the statistic and dictionary-based model only needs to count the frequency of each word. Given a set of training text, it takes no more than one minutes to train.
    b. It does not use much RAM and disk space. The only structure it needs to store in memory or disk is the Trie structure. Compared to the weights from machine learning, this structure saved many spaces.
    c. It has good accuracy. It uses Viterbi algorithm to find out the likelihood of the segmentation of the sentence, so it can overall find the one of the best choices if the sentence has ambiguous.

    The basic idea of this design is to find the likelihood of each combination of the word and output the one with the highest possibility of segmentation. Suppose we have the frequency of every word in the dictionary. In the simplest example, assume we have a string of characters $C_1C_2C_3$, and C_1C_2 can be a word and C_2C_3 can also be a word. So, we compare the P_1=Frequency(C_1C_2)*Frequency(C_3), P_2=Frequency(C_1)*Frequency(C_2C_3), and P_3= Frequency(C_1)* Frequency(C_2)* Frequency(C_3). Because each frequency is usually a very low number, P_3 in most time will not be the pick.

Now, we do a transformation on the frequency of the word. Because the frequency is usually a very low value (think about a word might appear 1 out of 1,000,000 of the corpus), it is very likely the product of the frequencies underflows. So, we take the negative log-likelihood of the frequency. The formula is given by:

$$C = -\log\left(\frac{\mathrm{f}}{\mathrm{N}}\right)$$

The f in the formula stand for the frequency of the word. N stands for the size of the corpus. Now, we sum up the weight instead of getting the product and pick the one with the lowest value.

ii. Enhancement on memory, caching

In the process of implementation of this model, there appears a problem that the memory consumed by the model is way higher than expected. After doing some research, I figured out it was caused by the programming language itself. PHP is a script language. The array it uses is a dynamic structure. It sacrifices some RAM for flexibility. [6] did a test on the PHP array and found that the size needed by PHP array is about 8 times larger the one used in C language. That means the more arrays you have in the code, the larger RAM it consumes. The Trie array became a problem because it is a structure of arrays of arrays. Here is a picture of Trie array.

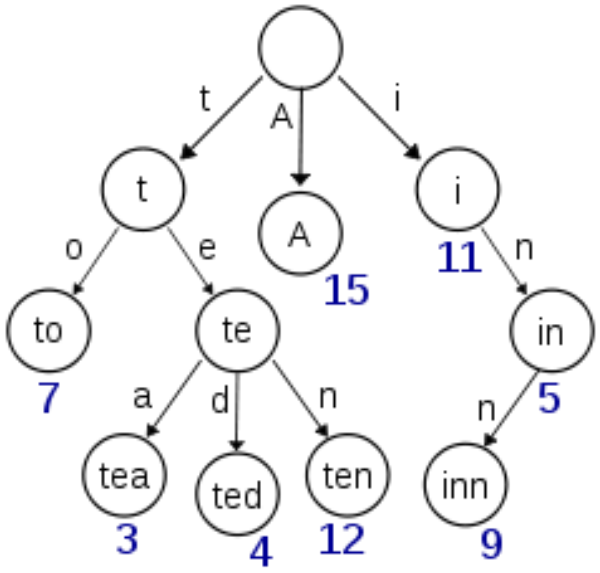

Figure 4: Trie Array

However, if the most part of the array is encoded in a format of string, then it will not consume much RAM. So, we can decode and only decode the part of it when needed to. This caused another problem. The decoding process takes too long. When we need to find a term, we need to decode before use. It turns out the segmenting process is 20-30 times slower than non-encoded one.

So, we take something in the middle: caching. Pareto principle (also known as the 80/20 rule) might be good description of the problem. The segmenter in most time will use a small partial of the whole data. We cach the entry of the word from the string and put it in caches when we encounter the word and discard the least frequent word from the cash. The caches have a very high hit rate. By this way, we can limit the RAM usage as well as the slow decoding process.

2. POS tagging

i. Introduction of the model

The model we use is the maximum entropy model. The reasons we consider this model are as follows:

a. The maximum entropy model is easier to implement. Yioop does not have third party library dependencies and will not have. And, different from other language like python, PHP does not have a machine learning library support. Implementing an easy enough multiclass classifier is necessary in my project.
b. Although the training phrase is very long, the classification process is quick.

We use the model like the tagger described in [Ratnaparkhi]. The model consists of the contexts and the possible tags from previous word. The model's joint probability of the context h and the tag t is defined as

$$\mathrm{p}(\mathrm{h},\mathrm{t}) = \pi\mu \prod_{j=1}^{k} a_j^{f_j^{(h,t)}}$$

π is a normalization constant. {μ, α_1, …, α_k} are weights and {f_1,…,f_k} are feature indicator functions. We take the negative log likelihood of this function and get:

$$\mathrm{p}'^{(\mathrm{h},\mathrm{t})} = -\log\big(\mathrm{p}(\mathrm{h},\mathrm{t})\big) = -\log(\pi\mu) + \sum_{j=1}^{k} f_j^{(h,t)} * (\log\big(-a_j\big))$$

We rewrite the formula to get:

$$\mathrm{p}'(\mathrm{h},\mathrm{t}) = \mathrm{w}_0 + \sum_{j=1}^{k} \mathrm{w}_j f_j^{(h,t)}$$

In the training phrase, we use cross entropy, to maximize the likelihood of the sigmoid function of p'.

$$\text{Sigmoid}(p') = \frac{1}{1+e^{-p'}}$$

Then

$$\text{Cross entropy loss} = -(y \log(\text{Sigmoid}(p')) + (1-y)\log(1-\text{Sigmoid}(p')))$$

We take the average of each cross-entropy loss for each w_i and train the data.

Suitable features are also an important part of this model. A feature should encode information to help to predict the POS tag. The limitations in this project are that we want to limit the usage of memory and disk spaces. So, we tried to limit the combination of the features and limit the number of classes of the tags.

Our model contains these features:

A. The current word (W_0)
B. The previous (next) two word (W_{-2}, W_{-1}, W_1, W_2)
C. The tag of the previous word (T_{-1}), and the tag of the character two before the current word (T_{-2})

The features above encode three types of the contexts. First one is the features based on the current and surrounding words. Given a word, the model will look at the current word as well as the precious two and next two words. For example, if the current word is 知识 ("knowledge"), it is very likely to be a noun because this word itself in most cases is a noun. In another example, 爱 ("love") can be a verb or noun.

The model will investigate the context of the text. If the previous word is 我 (“I”), this love is more likely to be a verb. Or, if the previous word is an adjective, then this love is more likely to be a noun.

To find the most likelihood of this tag, we find the min of the $p'(h,t)$ for all possible tags.

ii. Enhancement

This implementation also has problems that the weight of the file is too huge. Although there are not too many characters in Chinese, there are tons of words because of the combination. So, we sacrifice some accuracy to lower down the disk space for the trained data.

First enhancement is to pack the weight into two byte shot type. A floating number is 8 bytes in PHP, so we lose some precision in numbers.

Second enhancement is to discard some uncommon tags from the dataset. Some words have very special tags such as nt-abbre to indicate it is an abbreviation of a noun. Discarding those special tags does lower the accuracy of the model a little bit. However, it does decrease the size of the disk file for a huge amount.

3. NER

In the named entity recognition, I also used the maximum entropy model to classify the entities. Because I used the same model as POS tagging. So, I will skip the introduction of the model.

The difference between the POS tagging and named entity recognition is the different tags and features. Now, we need to predict what are the entities in a text. There are two type of encoding classes for the named entity. First one is called IO encoding. The other one is called IOB encoding. Here is an example:

	IO encoding	IOB encoding
Forrest	PER	B-PER
lived	O	O
in	O	O
Shanghai	LOC	B-LOC
Mainland	LOC	B-LOC
Of	LOC	I-LOC
China	LOC	I-LOC

IO encoding is called inside-outside encoding, it just checks whether the current word is or is not inside the entity and groups those together. However, Shanghai is one entity and Mainland of China is another. In IOB encoding, it also marks whether the word is the beginning of the entity. And now we can differ the two entities. In realities, the IOB encoding does not do an excellent job. In most of the time, it will just misclass the Mainland to be an I-LOC.

So, in my implementation, I used the IO encoding because it is faster, and it is not very common two person names or locations will next to each other.

Our model contains these features:

A. The current character (C_0)
B. The previous (next) two character (C_{-2}, C_{-1}, C_1, C_2)
C. The tag of the previous character (T_{-1}), and the tag of the character two before the current character (T_{-2})

Just as discussed in the POS tagging part, now we group the continuous same tags to be one entity.

4. Question and Answering system

Because of the time limitation, I did not adopt a new QA system for Yioop. I implemented the Triplet for Chinese sentence syntax.

The Triplet QA system consist of two part: indexing the knowledges and retrieving knowledges.

4.1. Indexing phrase

The first phrase is to index the knowledges. Knowledge-based system needs a huge number of knowledges for the system to work. Because Yioop is a search engine, it can crawl web documents. I implemented a rule-based Chinese sentence parser to extract the concept from documents. The concept here is a form of subject-predicate-object expression. It is also known as semantic triple. If the parser can parse the sentence into this format, then we substitute “qqq” into each of the entity and index it. For example, I have a sentence “鲍勃认识约翰” (“Bob knows John”). It can be parsed into triple 鲍勃-认识-约翰. So, we substitute qqq into it to get:

a. qqq 认识 约翰 (“qqq knows John”).
b. 鲍勃 qqq 约翰 (“Bob qqq John”).
c. 鲍勃 认识 qqq (“Bob knows qqq”).

We save these triplets into the databases for later use.

4.1.1 Parse tree generation

In order to generate the form of triplet, we need to parse the sentence into the parse tree. The First step is to segment the sentence and tag the part-of-speech of the words. These two steps are accomplished with precious work.

Most Chinese sentence follow the form of subject-predicate-object. There are exceptions. For example, we can just move topic item to the front.

	Example 1	Example2
Chinese:	你 饭 都 吃 好了。	去死 吧 你!
Words in English:	You meal already ate yet.	Go to hell you!
Translation:	You have eaten meal already.	You go to hell!

The exceptions are not parse-able in my parser, otherwise it can parse most sentence.

It is very common that one Chinese sentence contains multiple sub-sentences. If the sub-sentences share the same topic or subject, they can get combined into one. The below is an example is from Wikipedia:

珠穆朗玛峰是喜马拉雅山脉的主峰，同时是世界海拔最高的山峰，位于中国与尼泊尔边境线上。

Translation:

Mount Qomolangma is the main peak of the Himalayas, at the same time is the highest mountain in the world, is located on the border between China and Nepal.

You can see this sentence consists of tree sub-sentences. And, the sub-sentences, except the first one, do not have a subject because all sub-sentences share the same subject.

Another Chinese syntax that is different from English is that Chinese words consist of many particle words. Those words, such as 的, 地, 得, 了, 着, do not have meaning by themselves. Sometimes they even have no meaning in the text. Chinese people just feel better to include those words. Some particle words are strongly associated with tense, so it is not good just to discard them.

Figure 5 shows the general parse tree generated by the system.

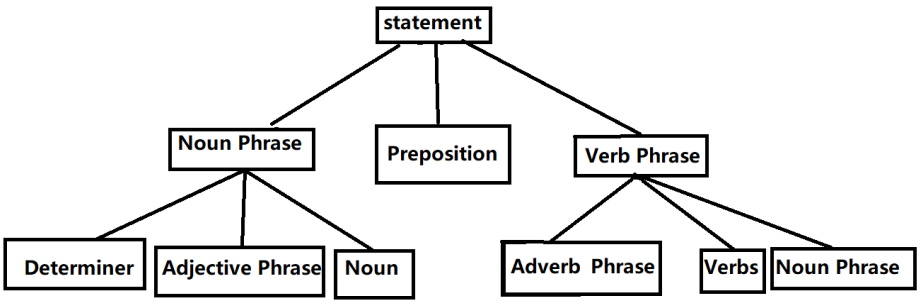

Figure 5: Chinese Gramma tree Parser

4.2. Retrieve the knowledge

Our system will do similar steps in the indexing phrase when getting the query. If the query is a question, then it investigates the index and calculate the best answer and return to user.

Experimental Results

1. Chinese Segmentation

The segmenter was tested under two datasets: Academia Sinica (AS) dataset Peking University (PKU). The accuracy for each test dataset is:

Dataset	Stochastic Segmentation (ours)	Reverse Max Match	Neural with Multi-Criteria Learning (Current Highest)
AS	94.3%	89.42%	96.6%
PKU	86.8%	83.10%	96.6%

The Stochastic Segmenter improves the system by 7.5% and 6% on AS and PKU dataset.

Figure 6: Chinese Segmentation Example

For the enhancement, the memory usage vs running time. The more memory it uses, the less time it does the work.

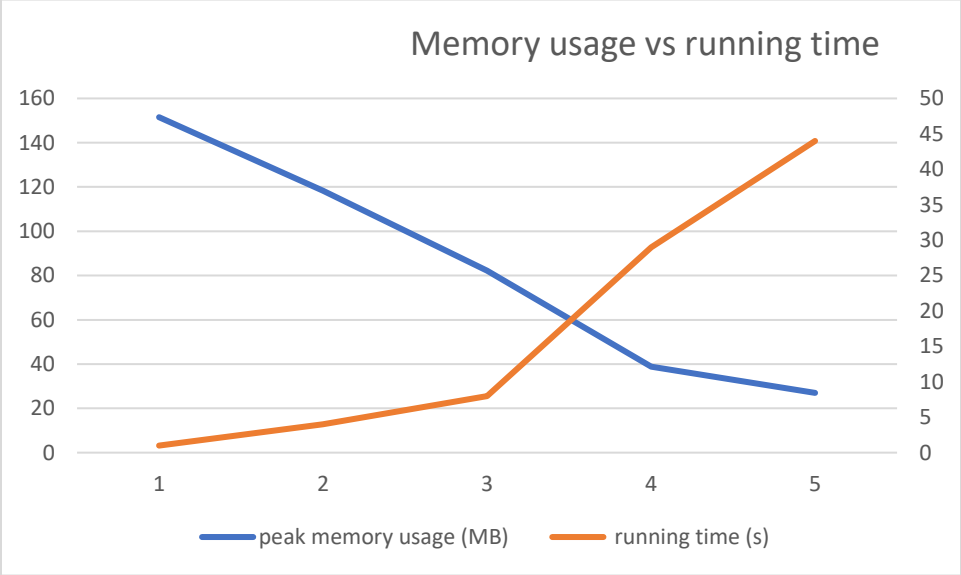

Figure 7: Memory Usage vs. Running Time

2. POS tagging

The POS tagger was tested using Chinese Three Bank. Because the dataset is very huge, I did not train all of them. I used 30% of the data for train and 10% of the data for test.

The first part of the result is tested without the limitation of the memory and size of the storage file.

After it ran about 1500 epochs, it reached the highest accuracy, which is 92.5%. Then it starts to decrease because of the over fitting.

The second part of the result is tested with the limitation of the memory and the size of the storage. With decreased tags size and weight precision, after it ran about 1500 epochs, it reached the highest accuracy, which is 89.5%.

Here is a picture of epoch versus accuracy for the one without the limitation:

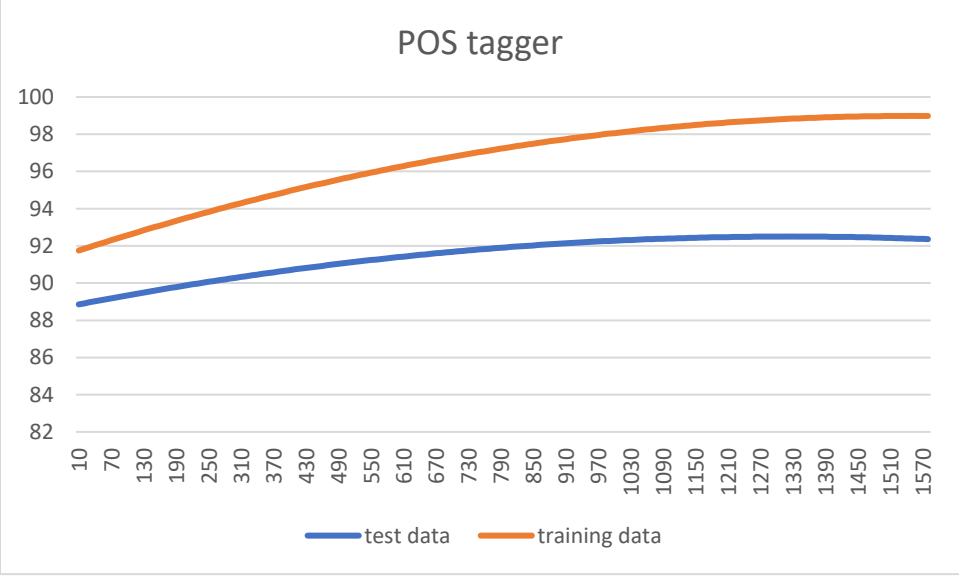

Figure 8: Accuracy for test data and training data

Here is a figure of the POS tagging result:

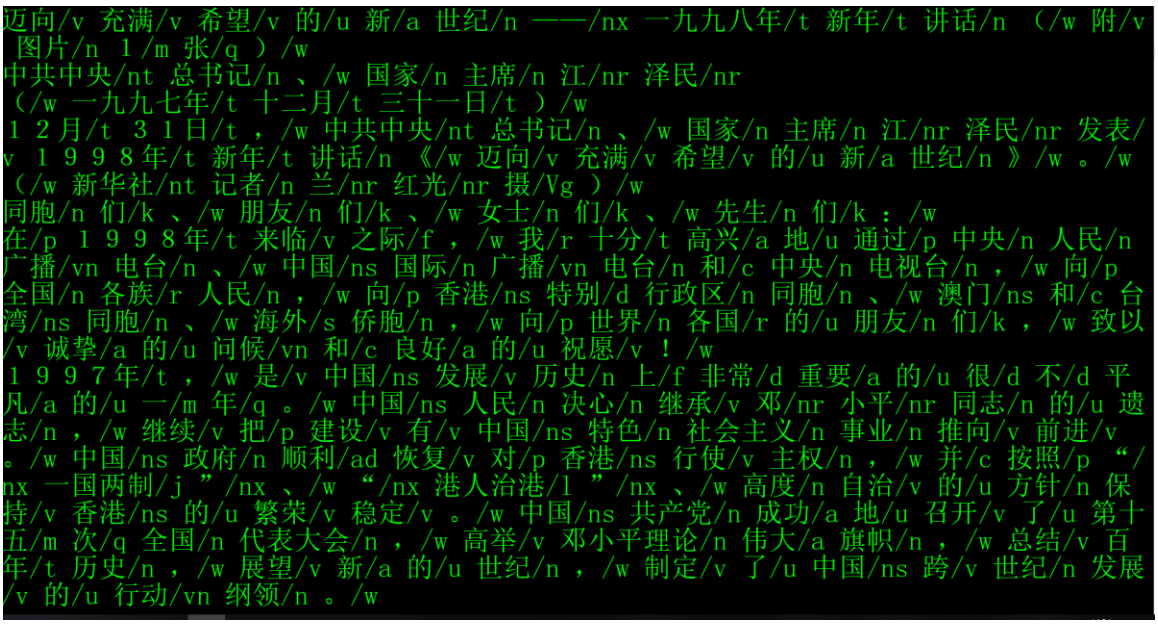

Figure 9: Chinese POS tagging

3. Named Entity Recognition

The named entity recognizer was trained and tested under MSRA dataset. The accuracy for the test data is around 83.6%. From the test data observed, the location and the person name are more likely to be tagged correctly. However, the organization names are less likely to be tagged correctly. The reason is that the location names are very stable. There are no new introduced names. Person names consist of some common name characters, so they are more likely to hit. However, the organization names consist of either location characters or widely used characters, so they are unlikely to be tagged correctly.

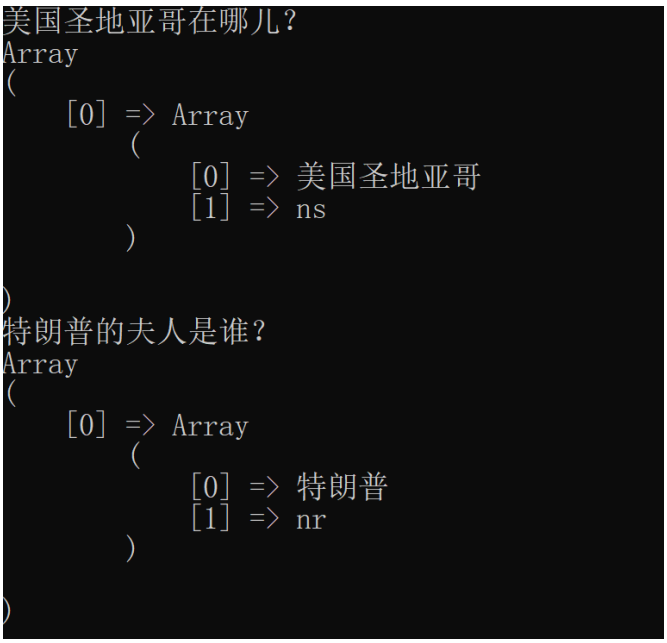

Figure 10: Named Entity Recognition

4. Question and answering System

4.1 Tree Parser Testcase

In this section, we test the accuracy of the parser successfully parse the text into semantic triple.

Figure 4 shows the triplets extracted from the sentence:

珠穆朗玛峰是喜马拉雅山脉的主峰，同时是世界海拔最高的山峰，位于中国与尼泊尔边境线上。

Translation:

Mount Everest is the main peak of the Himalayas. At the same time, it is the highest mountain in the world. It is located on the border between China and Nepal.

```
[QUESTION_ANSWER_LIST] => Array
    (
        [qqq 是 主峰] => 珠穆朗玛峰
        [珠穆朗玛峰 qqq 主峰] => 是
        [珠穆朗玛峰 是 qqq] => 世界最高峰
        [qqq 是 喜马拉雅山脉 的 主峰] => 珠穆朗玛峰
        [珠穆朗玛峰 qqq 喜马拉雅山脉 的 主峰] => 是
        [qqq 是 世界海拔] => 珠穆朗玛峰
        [珠穆朗玛峰 qqq 世界海拔] => 是
        [qqq 是 世界海拔  最高 的 山峰] => 珠穆朗玛峰
        [珠穆朗玛峰 qqq 世界海拔  最高 的 山峰] => 是
        [qqq 位于 中国与尼泊尔边境线上] => 珠穆朗玛峰
        [珠穆朗玛峰 qqq 中国与尼泊尔边境线上] => 位于
        [珠穆朗玛峰 位于 qqq] => 中国与尼泊尔边境线上
        [qqq 是 世界最高峰] => 珠穆朗玛峰
        [珠穆朗玛峰 qqq 世界最高峰] => 是
    )
```

Figure 11: Triplets of Chinese sentence

The parser tree extracts triplets from all three sub-sentence information contained in this sentence.

Conclusion and Future work

Natural Language Processing is a very hard area. Completing a series of tasks in this project gives me a basic idea how machine can “understand” human language.

In this project, we improved existing Chinese Segmentation Algorithm. We implemented a POS tagger for Chinese words. We implemented a named entity recognizer for Chinese. Finally, we implemented a Chinese Question Answering System. Now the system can process Chinese text at a better accuracy and answer simple factoid questions.

Many different approaches are used in this project, including rule-based for question answering system, statistical based for Chinese word Segmentation. Machine learning approaches for Part-of-Speech Tagging and Named Entity Recognition.

There are many challenges in this project. The Major challenge is to implement a maximum entropy model. PHP does not have internal Machine Learning Library. It took me some time to learn and implement the machine learning techniques. Another challenge I had is to parse Chinese gramma. The Chinese gramma is more complex than I thought. Even I speak Chinese in my whole life, I still had hard time to generate the Chinese Gramma Parser.

There are many futures works can be done. The knowledge-based Question Answering system is less popular nowadays because it can only answer limited factoid questions. The IR-based or hybrid QA systems can answer more questions based on the all sources from internet. Yioop is a search engine so it will have this system in the future.

Reference

[1] Sproat, Richard & Shih, Chilin & Gale, William & Chang, Nancy. (2002). A Stochastic Finite-State Word-Segmentation Algorithm For Chinese. Computational Linguistics. 22. 10.3115/981732.981742.

[2] Huihsin Tseng, Daniel Jurafsky, and Christopher Manning. 2005. Morphological features help pos tagging of unknown words across language varieties. In Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing.

[3] Malouf, R (2002). A comparison of algorithms for maximum entropy parameter estimation (PDF). Sixth Conf. on Natural Language Learning (CoNLL). pp. 49–55.

[4] Prager, J. Information Retrieval, 2006 Open-Domain Question-Answering

[5] Fei Xia, The Part-Of-Speech Tagging Guidelines for the Penn Chinese Treebank (3.0) https://repository.upenn.edu/cgi/viewcontent.cgi?article=1039&context=ircs_reports

[6] Popov, N. How big are PHP arrays (and values) really? https://nikic.github.io/2011/12/12/How-big-are-PHP-arrays-really-Hint-BIG.html

[7] Pollett, C. "Open Source Search Engine Software!" Open Source Search Engine Software. https://www.seekquarry.com/