FudanNLP: A Toolkit for Chinese Natural Language Processing

Xipeng Qiu, Qi Zhang, Xuanjing Huang
Fudan University, 825 Zhangheng Road, Shanghai, China
xpqiu@fudan.edu.cn, qz@fudan.edu.cn, xjhuang@fudan.edu.cn

Abstract

The growing need for Chinese natural language processing (NLP) is largely in a range of research and commercial applications. However, most of the currently Chinese NLP tools or components still have a wide range of issues need to be further improved and developed. FudanNLP is an open source toolkit for Chinese natural language processing (NLP), which uses statistics-based and rule-based methods to deal with Chinese NLP tasks, such as word segmentation, part-of-speech tagging, named entity recognition, dependency parsing, time phrase recognition, anaphora resolution and so on.

1 Introduction

Chinese is one of the most widely used languages in this world, and the proportion that Chinese language holds on the Internet is also quite high. Under the current circumstances, there are greater and greater demands for intelligent processing and analyzing of the Chinese texts.

Similar to English, the main tasks in Chinese NLP include word segmentation (CWS), part-of-speech (POS) tagging, named entity recognition (NER), syntactic parsing, anaphora resolution (AR), and so on. Although the general ways are essentially the same for English and Chinese, the implementation details are different. It is also non-trivial to optimize these methods for Chinese NLP tasks.

There are also some toolkits to be used for NLP, such as Stanford CoreNLP[1], Apache OpenNLP[2], Curator[3] and NLTK[4]. But these toolkits are developed mainly for English and not optimized for Chinese.

In order to customize an optimized system for Chinese language process, we implement an open source toolkit, FudanNLP[5], which is written in Java. Since most of the state-of-the-art methods for NLP are based on statistical learning, the whole framework of our toolkit is established around statistics-based methods, supplemented by some rule-based methods. Therefore, the quality of training data is crucial for our toolkit. However, we find that there are some drawbacks in currently most commonly used corpora, such as CTB (Xia, 2000) and CoNLL (Hajič et al., 2009) corpora. For example, in CTB corpus, the set of POS tags is relative small and some categories are derived from the perspective of English grammar. And in CoNLL corpus, the head words are often interrogative particles and punctuations, which are unidiomatic in Chinese. These drawbacks bring more challenges to further analyses, such as information extraction and semantic understanding. Therefore, we first construct a corpus with a modified guideline, which is more in accordance with the common understanding for Chinese grammar.

In addition to the basic Chinese NLP tasks

[1] `http://nlp.stanford.edu/software/corenlp.shtml`

[2] `http://incubator.apache.org/opennlp/`

[3] `http://cogcomp.cs.illinois.edu/page/software_view/Curator`

[4] `http://www.nltk.org/`

[5] `http://fudannlp.googlecode.com`

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 49–54,
Sofia, Bulgaria, August 4-9 2013. ©2013 Association for Computational Linguistics

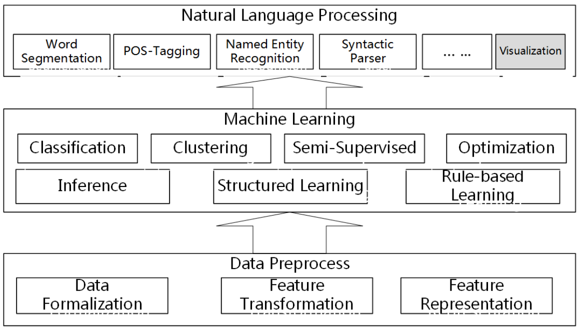


Figure 1: System Structure of FudanNLP

mentioned above, the toolkit also provides many minor functions, such as text classification, dependency tree kernel, tree pattern-based information extraction, keywords extraction, translation between simplified and traditional Chinese, and so on.

Currently, our toolkit has been used by many universities and companies for various applications, such as the dialogue system, social computing, recommendation system and vertical search.

The rest of the demonstration is organized as follows. We first briefly describe our system and its main components in section 2. Then we show system performances in section 3. Section 4 introduces three ways to use our toolkit. In section 5, we summarize the paper and give some directions for our future efforts.

2 System Overview

The components of our system have three layers of structure: data preprocessing, machine learning and natural language processing, which is shown in Figure 1. We will introduce these components in detail in the following subsections.

2.1 Data Preprocessing Component

In the natural language processing system, the original input is always text. However, the statistical machine learning methods often deal with data with vector-based representation. So we firstly need to preprocess the input texts and transform them to the required format. Due to the fact that text data is usually discrete and sparse, the sparse vector structure is largely used. Similar to Mallet (McCallum, 2002), we use the pipeline structure for a flexible transformation of various data. The pipeline consists of several serial or parallel modules. Each module, called "pipe", is aimed at a single and simple function.

For example, when we transform a sentence into a vector with "bag-of-words", the transformation process would involve the following serial pipes:

1. String2Token Pipe: to transform a string into word tokens.

2. Token2Index Pipe: to look up the word alphabet to get the indices of the words.

3. WeightByFrequency Pipe: to calculate the vector weight for each word according to its frequency of occurrence.

With the pipeline structure, the data preprocessing component has good flexibility, extensibility and reusability.

2.2 Machine Learning Component

The outputs of NLP are often structured, so the structured learning is our core module. Structured learning is the task of assigning a structured label $\mathbf{y}$ to an input $\mathbf{x}$. The label $\mathbf{y}$ can be a discrete variable, a sequence, a tree or a more complex structure.

To illustrate by a sample $\mathbf{x}$, we define the feature as $\Phi(\mathbf{x}, \mathbf{y})$. Thus, we can label $\mathbf{x}$ with a score function,

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} F(\mathbf{w}, \Phi(\mathbf{x}, \mathbf{y})), \tag{1}$$

where $\mathbf{w}$ is the parameter of function $F(\cdot)$. The feature vector $\Phi(\mathbf{x}, \mathbf{y})$ consists of lots of overlapping features, which is the chief benefit of a discriminative model.

For example, in sequence labeling, both $\mathbf{x} = x_1, \ldots, x_L$ and $\mathbf{y} = y_1, \ldots, y_L$ are sequences. For first-order Markov sequence labeling, the feature can be denoted as $\phi_k(y_{i-1}, y_i, \mathbf{x}, i)$, where i is the position in the sequence. Then the score function can be rewritten as

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} F(\sum_{i=1}^{L} \sum_{k} w_k \phi_k(y_{i-1}, y_i, \mathbf{x}, i)), \tag{2}$$

where L is the length of $\mathbf{x}$.

Different algorithms vary in the definition of $F(\cdot)$ and the corresponding objective function.

$F(\cdot)$ is usually defined as a linear or exponential family function. For example, in conditional random fields (CRFs) (Lafferty et al., 2001), $F(\cdot)$ is defined as:

$$P_{\mathbf{w}}(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_{\mathbf{w}}} \exp(\mathbf{w}^T \Phi(\mathbf{x}, \mathbf{y})), \qquad (3)$$

where $Z_{\mathbf{w}}$ is the normalization constant such that it makes the sum of all the terms one.

In FudanNLP, the linear function is universally used as the objective function. Eq. (1) is written as:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} < \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) > . \qquad (4)$$

2.2.1 Training

In the training stage, we use the passive-aggressive algorithm to learn the model parameters. Passive-aggressive (PA) algorithm (Crammer et al., 2006) was proposed for normal multi-class classification and can be easily extended to structure learning (Crammer et al., 2005). Like Perceptron, PA is an online learning algorithm.

2.2.2 Inference

For consistency with statistical machine learning, we call the process to calculate the Eq.(1) as "inference". In structured learning, the number of possible solutions is very huge, so dynamic programming or approximate approaches are often used for efficiency. For NLP tasks, the most popular structure is sequence. To label the sequence, we use Viterbi dynamic programming to solve the inference problem in Eq. (4).

Our system can support any order of Viterbi decoding. In addition, we also implement a constrained Viterbi algorithm to reduce the number of possible solutions by pre-defined rules. For example, when we know the probable labels, we delete the unreachable states from state transition matrix. It is very useful for CWS and POS tagging with sequence labeling. When we have a word dictionary or know the POS for some words, we can get more accurate results.

2.2.3 Other Algorithms

Apart from the core modules of structured learning, our system also includes several traditional machine learning algorithms, such as Perceptron, Adaboost, kNN, k-means, and so on.

2.3 Natural Language Processing Components

Our toolkit provides the basic NLP functions, such as word segmentation, part-of-speech tagging, named entity recognition, syntactic parsing, temporal phrase recognition, anaphora resolution, and so on. These functions are trained on our developed corpus. We also develop a visualization module to displaying the output. Table 1 shows the output representation of our toolkit.

2.3.1 Chinese Word Segmentation

Different from English, Chinese sentences are written in a continuous sequence of characters without explicit delimiters such as the blank space. Since the meanings of most Chinese characters are not complete, words are the basic syntactic and semantic units. Therefore, it is indispensable step to segment the sentence into words in Chinese language processing.

We use character-based sequence labeling (Peng et al., 2004) to find the boundaries of words. Besides the carefully chosen features, we also use the meaning of character drawn from HowNet(Dong and Dong, 2006), which improves the performance greatly. Since unknown words detection is still one of main challenges of Chinese word segmentation. We implement a constrained Viterbi algorithm to allow users to add their own word dictionary.

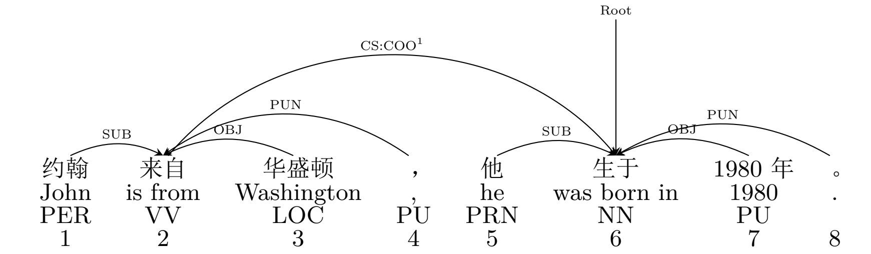
2.3.2 POS tagging

Chinese POS tagging is very different from that in English. There are no morphological changes for a word among its different POS tags. Therefore, most of Chinese words may have multiple POS tags. For example, there are different morphologies in English for the word "毁灭 (destroy)", such as "destroyed", "destroying" and "destruction". But in Chinese, there is just one same form(Xia, 2000).

There are two popular guidelines to tag the word's POS: CTB (Xia, 2000) and PKU (Yu et al., 2001). We take into account both the weaknesses and the strengths of these two guidelines, and propose our guideline for better subsequent analyses, such as parser and named entity recognition. For example, the proper name is labeled as "NR" in CTB, while we label it with one of four categories: person,

Input:
约翰来自华盛顿，他生于 1980 年。
John is from Washington, and he was born in 1980.

Output:



NER:		
1	→	PER
3	→	LOC
AR:		
5	→	1
TIME:		
7	→	1980

[1] CS:COO means the coordinate complex sentence.

Table 1: Example of the output representation of our toolkit

location, organization and other proper name. Conversely, we merge the “VC” and “VE” into “VV” since there is no link verb in Chinese. Finally, we use a tag set with 39 categories in total.

Since a POS tag is assigned to each word, not to each character, Chinese POS tagging has two ways: pipeline method or joint method. Currently, the joint method is more popular and effective because it uses more flexible features and can reduce the error propagation (Ng and Low, 2004). In our system, we implement both methods for POS tagging. Besides, we also use some knowledge to improve the performance, such as Chinese surname and the common suffixes of the names of locations and organizations.

2.3.3 Named Entity Recognition

In Chinese named entity recognition (NER), there are usually three kinds of named entities (NEs) to be dealt with: names of persons (PER) , locations (LOC) and organizations (ORG). Unlike English, there is no obvious identification for NEs, such as initial capitals. The internal structures are also different for different kinds of NEs, so it is difficult to build a unified model for named entity recognition.

Our NER is based on the results of POS tagging and uses some customize features to detect NEs. First, the number of NEs is very large and the new NEs are endlessly emerging, so it is impossible to store them in dictionary. Since the internal structures are relatively more important, we use language models to capture the internal structures. Second, we merge the continuous NEs with some rule-based strategies. For example, we combine the continuous words “人民/NN 大会堂/NN” into “ 人民大会堂/LOC”.

2.3.4 Dependency parsing

Our syntactic parser is currently a dependency parser, which is implemented with the shift-reduce deterministic algorithm based on the work in (Yamada and Matsumoto, 2003). The syntactic structure of Chinese is more complex than that of English, and semantic meaning is more dominant than syntax in Chinese sentences. So we select the dependency parser to avoid the minutiae in syntactic constituents and wish to pay more attention to the subsequent semantic analysis. Since the structure of the Chinese language is quite different from that of English, we use more effective features according to the characteristics of Chinese sentences.

The common used corpus for Chinese dependency parsing is CoNLL corpus (Hajič et al., 2009). However, there are some illogical cases in CoNLL corpus. For example, the head words are often interrogative particles and punctuations. Our guideline is based on common understanding for Chinese grammar. The Chinese syntactic components usually include subject, predicate, object, attribute, adverbial modifier and complement. Figure 2 and 3 show the differences between the trees of CoNLL and our Corpus. Table 2 shows some primary dependency relations in our guideline.

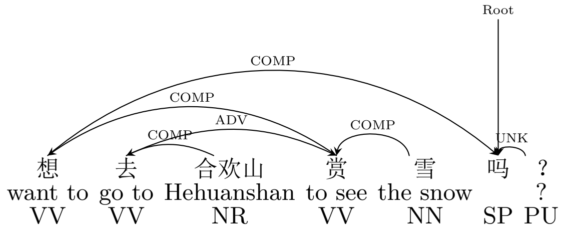


Figure 2: Dependency Tree in CoNLL Corpus

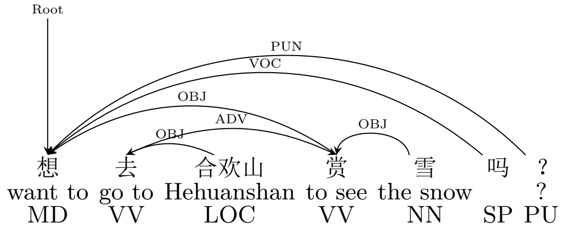


Figure 3: Dependency Tree in Our Corpus

Relations	Chinese	Definitions
SUB	主语	Subject
PRED	谓语	Predicate
OBJ	宾语	Object
ATT	定语	Attribute
ADV	状语	Adverbial Modifier
COMP	补语	Complement
SVP	连动	Serial Verb Phrases
SUB-OBJ	兼语	Pivotal Construction
VOC	语态	Voice
TEN	时态	Tense
PUN	标点	Punctuation

Table 2: Some primary dependency relations

2.3.5 Temporal Phrase Recognition and Normalization

Chinese temporal phrases is more flexible than English. Firstly, there are two calendars: Gregorian and lunar calendars. Both of them are frequently used. Secondly, the forms of same temporal phrase are various, which often consists of Chinese characters, Arabic numerals and English letters, such as "早上 10 点" and "10:00 PM".

Different from the general process based on machine learning, we implement the time phrase recognizer with a rule-based method. These rules include 376 regular expressions and nearly a hundred logical judgments.

After recognizing the temporal phrases, we normalize them with a standard time format. For a phrase indicating a relative time , such as "一年后" and " 一小时后", we first find the base time in the context. If no base time is found, or there is also no temporal phrase to indicate the base time (such as "明天"), we set the base time to the current system time. Table 3 gives examples for our temporal phrase recognition module.

<table>
<tr><td colspan="2">Input:
08 年北京举行奥运会，8 月 8 号开幕。四年后的七月二十七日，伦敦奥运开幕。
The Beijing Olympic Games took place from August 8, 2008. Four years later, the London Olympic Games took place from July 21.

今天我很忙，晚上 9 点才能下班。周日也要加班。
I'm busy today, and have to come off duty after 9:00 PM. And I also have to work this Sunday.</td></tr>
<tr><td colspan="2">Output:</td></tr>
<tr><td>08 年 (2008)
8 月 8 号 (August 8)
七月二十七日 (July 21)
今天 (today)
晚上 9 点 (9:00 PM)
周日 (this Sunday)</td><td>2008
2008-8-8
2012-7-27
2012-2-22[1]
2012-2-22 21:00
2012-2-26</td></tr>
</table>

[1] The base time is 2012-02-22 10:00AM.

Table 3: Examples for Temporal Phrase Recognition

2.3.6 Anaphora Resolution

Anaphora resolution is to detect the pronouns and find what they are referring to. We first find all pronouns and entity names, then use a classifier to predict whether there is a relation between each pair of pronoun and entity name. Table 4 gives examples for our anaphora resolution module.

<table>
<tr><td colspan="2">Input:
牛津大学创建于 1167 年。它位于英国牛津城。这个大学培育了好多优秀的学生。
Oxford University is founded in 1167. It is located in Oxford, UK. The university has nurtured a lot of good students.</td></tr>
<tr><td>Output:</td><td></td></tr>
<tr><td>它 (It)
这个大学 (The university)</td><td>牛津大学
牛津大学 (Oxford University)</td></tr>
</table>

Table 4: Examples for Anaphora Resolution

3 System Performances

In this section, we investigate the performances for the six tasks: Chinese word segmentation (CWS), POS tagging (POS), named entity recognition (NER) and dependency parser(DePar), Temporal Phrase Recognition (TPR) and Anaphora Resolution (AR). We use 5-fold cross validation on our developed corpus. The corpus includes 65,745 sentences and 959,846 words. The performances are shown in Table 5.

Task	Accuracy	Speed[1]	Memory
CWS	97.5%	98.9K	66M
POS	93.4%	44.5K	110M
NER	98.40%	38K	30M
DePar	85.3%	21.1	80M
TPR	95.16%	22.9k	237K
AR	70.3%	35.7K	52K

[1] characters per second. Test environment: CPU 2.67GHz, JRE 7.

Table 5: System Performances

4 Usages

We provide three ways to use our toolkit.

Firstly, our toolkit can be used as library. Users can call application programming interfaces (API) in their own applications.

Secondly, users can also invoke the main NLP modules to process the inputs (strings or files) from the command line directly.

Thirdly, the web services are provided for platform-independent and language-independent use. We use a REST (Representational State Transfer) architecture, in which the web services are viewed as resources and can be identified by their URLs.

5 Conclusions

In this demonstration, we have described the system, FudanNLP, which is a Java-based open source toolkit for Chinese natural language processing. In the future, we will add more functions, such as semantic parsing. Besides, we will also optimize the algorithms and codes to improve the system performances.

Acknowledgments

We would like to thank all the people[6] involved with our FudanNLP project. This work was funded by NSFC (No.61003091 and No.61073069) and 973 Program (No.2010CB327900).

References

K. Crammer, R. McDonald, and F. Pereira. 2005. Scalable large-margin online learning for structured classification. In *NIPS Workshop on Learning With Structured Outputs*. Citeseer.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585.

Z. Dong and Q. Dong. 2006. *Hownet And the Computation of Meaning*. World Scientific Publishing Co., Inc. River Edge, NJ, USA.

J. Hajič, M. Ciaramita, R. Johansson, D. Kawahara, M.A. Martí, L. Màrquez, A. Meyers, J. Nivre, S. Padó, J. Štěpánek, et al. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–18. Association for Computational Linguistics.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

H.T. Ng and J.K. Low. 2004. Chinese part-of-speech tagging: one-at-a-time or all-at-once? word-based or character-based. In *Proceedings of EMNLP*, volume 4.

F. Peng, F. Feng, and A. McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. *Proceedings of the 20th international conference on Computational Linguistics*.

F. Xia, 2000. *The part-of-speech tagging guidelines for the penn chinese treebank (3.0)*.

H. Yamada and Y. Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of the International Workshop on Parsing Technologies (IWPT)*, volume 3.

S. Yu, J. Lu, X. Zhu, H. Duan, S. Kang, H. Sun, H. Wang, Q. Zhao, and W. Zhan. 2001. Processing norms of modern chinese corpus. Technical report, Technical report.

[6] `https://code.google.com/p/fudannlp/wiki/People`