LTP: A Chinese Language Technology Platform

Wanxiang Che, Zhenghua Li, Ting Liu
Research Center for Information Retrieval
MOE-Microsoft Key Laboratory of Natural Language Processing and Speech
School of Computer Science and Technology
Harbin Institute of Technology
`{car, lzh, tliu}@ir.hit.edu.cn`

Abstract

LTP (Language Technology Platform) is an integrated Chinese processing platform which includes a suite of high performance natural language processing (NLP) modules and relevant corpora. Especially for the syntactic and semantic parsing modules, we achieved good results in some relevant evaluations, such as CoNLL and SemEval. Based on XML internal data representation, users can easily use these modules and corpora by invoking DLL (Dynamic Link Library) or Web service APIs (Application Program Interface), and view the processing results directly by the visualization tool.

1 Introduction

A Chinese natural language processing (NLP) platform always includes lexical analysis (word segmentation, part-of-speech tagging, named entity recognition), syntactic parsing and semantic parsing (word sense disambiguation, semantic role labeling) modules. It is a laborious and time-consuming work for researchers to develop a full NLP platform, especially for Chinese, which has fewer existing NLP tools. Therefore, it should be of particular concern to build an integrated Chinese processing platform. There are some key problems for such a platform: providing high performance language processing modules, integrating these modules smoothly, using processing results conveniently, and showing processing results directly.

LTP (Language Technology Platform), a Chinese processing platform, is built to solve the above mentioned problems. It uses XML to transfer data through modules and provides all sorts of high performance Chinese processing modules, some DLL or Web service APIs, visualization tools, and some relevant corpora.

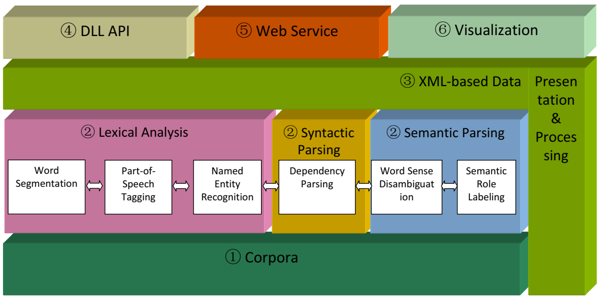


Figure 1: The architecture of LTP

2 Language Technology Platform

LTP (Language Technology Platform)[1] is an integrated Chinese processing platform. Its architecture is shown in Figure 1. From bottom to up, LTP comprises 6 components: ① Corpora, ② Various Chinese processing modules, ③ XML based internal data presentation and processing, ④ DLL API, ⑤ Web service, and ⑥ Visualization tool. In the following sections, we will introduce these components in detail.

2.1 Corpora

Many NLP tasks are based on annotated corpora. We distributed two key corpora used by LTP.

First, WordMap is a Chinese thesaurus which contains 100,093 words. In WordMap, each word sense belongs to a five-level categories. There are 12 top, about 100 second and 1,500 third level, and more fourth and fifth level categories. For instance, the Chinese word "材料" has the following two senses:

[1] http://ir.hit.edu.cn/ltp/

Coling 2010: Demonstration Volume, pages 13–16,
Beijing, August 2010

1. “物(entity) → 统称(common name) → 物资(goods) → 物资(goods) → 材料(material)”

2. “人(human beings) → 才识(ability) → 俊杰(hero) → 人才(talents) → 人才(talents)”

We can see that the two senses belong to “物” (entity) and “人” (human beings) top categories respectively. In each category, the concept becomes more and more specifical.

The second corpus is Chinese Dependency Treebank (CDT) (Liu et al., 2006). It is annotated with the dependency structure and contains 24 dependency relation tags, such as SUB, OBJ, and ADV. It consists of 10,000 sentences randomly extracted from the first six-month corpus of People’s Daily (China) in 1998, which has been annotated with lexical tags, including word segmentation, part-of-speech tagging, and named entity recognition tags[2].

2.2 Chinese Processing Modules

We have developed 6 state-of-the-art Chinese processing modules for LTP.

1. Word Segmentation (WordSeg): A CRF model (Lafferty et al., 2001) is used to segment Chinese words. All of the People’s Daily (China) corpus is used as training data.

2. Part-of-Speech Tagging (POSTag): We adopt SVMTool[3] for Chinese POS tagging task (Wang et al., 2009). The People’s Daily corpus is also used here.

3. Named Entity Recognition (NER): LTP can identify six sorts of named entity: Person, Loc, Org, Time, Date and Quantity. A maximum entropy model (Berger et al., 1996) is adopted here. We still used the People’s Daily corpus.

4. Word Sense Disambiguation (WSD): This is an all word WSD system, which labels the WordMap sense of each word. It adopts an SVM model (Guo et al., 2007), which obtains the best performance in SemEval 2009 Task 11: English Lexical Sample Task via English-Chinese Parallel Text.

5. Syntactic Parsing (Parser): Dependency grammar is used in our syntactic parser. A high order graph-based model (Che et al., 2009) is adopted here which achieved the third place of the dependency syntactic parsing subtask in the CoNLL-2009 Syntactic and Semantic Dependencies in Multiple Languages Shared Task (Hajič et al., 2009).

Modules	Performance	Speed
WordSeg	$F1 = 97.4$	185KB/s
POSTag	The overall $Accuracy = 97.80\%$, and the out of vocabulary word $Accuracy = 85.48\%$	56.3KB/s
NER	The overall $F1 = 92.25$	14.4KB/s
WSD	The all word WSD $Accuracy = 94.34\%$ and the multi-sense word $Accuracy = 91.29\%$	7.2KB/s
Parser	LAS (Labeled Attachment Score) = 73.91% and UAS (Unlabeled Attachment Score) = 78.23%	0.2KB/s
SRL	$F1 = 77.15$	1.3KB/s

Table 1: The performance and speed for each module.

6. Semantic Role Labeling (SRL): SRL is to identify the relations between predicates in a sentence and their associated arguments. The module is based on syntactic parser. A maximum entropy model (Che et al., 2009) is adopted here which achieved the **first place** in the joint task of syntactic and semantic dependencies of the CoNLL-2009 Shared Task.

Table 1 shows the performance and speed of each module in detail. The performances are obtained with n-fold cross-validation method. The speed is gotten on a machine with Xeon 2.0GHz CPU and 4G Memory.

At present, LTP processes these modules with a cascaded mechanism, i.e., some higher-level processing modules depend on other lower-level modules. For example, WSD needs to take the output of POSTag as input; while before POSTag, the document must be processed with WordSeg. LTP can guarantee that the lower-level modules are invoked automatically when invoking higher-level modules.

2.3 LTML

We adopt eXtensible Markup Language (XML) as the internal data presentation for some reasons. First, XML is a simple, flexible text format, and plays an increasingly important role in the ex-

[2] http://icl.pku.edu.cn/icl_res/

[3] http://www.lsi.upc.edu/∼nlp/SVMTool/

change of a wide variety of data on the Web and elsewhere. Second, there exist many powerful and simple XML parsers. With these tools, we can easily and effectively achieve all kinds of operations on XML. Finally, based on XML, we can easily implement visualization with some script languages such as JavaScript.

Based on XML, we have designed a tag-set for NLP platform, named LTML (Language Technology Markup Language). Basically, we regard a word as a unit. The word has attributes such as id, pos, wsd, etc., which indicate the index, part-of-speech, word sense, etc. information of the word. A sentence consists of a word sequence and then a series of sentences compose a paragraph. The semantic role labeling arguments are attached to semantic predicate words. The meaning of each tag and attribute are explained in Table 2.

Tag	Meaning	Attr.	Meaning
<ltml>	Root node		
<doc>	Document level		
<para>	Paragraph in doc	id	Paragraph index in doc
<sent>	Sentence in para	id	Sentence index in paragraph
<word>	Word in sentence	id	Word index in sentence
		cont	Word content
		pos	Part of speech of word
		ne	Named entity type of word
		wsd	Word sense code in WordMap
		parent	Word id of this word depends on in syntax tree
		relate	Syntax relation type
<arg>	Semantic arguments of a word	id	Argument index of this word
		type	Semantic role of this argument
		beg	Beginning word id of this argument
		end	Ending word id of this argument

Table 2: Tags and attributes of LTML

2.4 DLL API

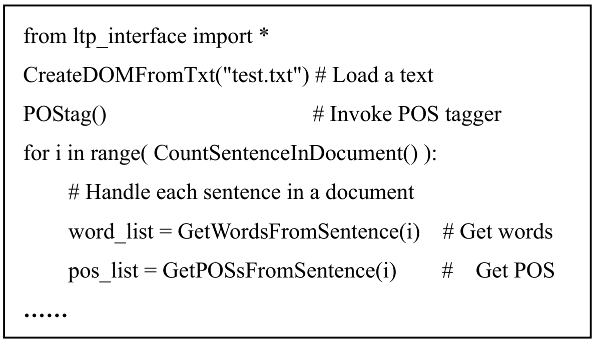
In order to gain the analysis results of LTP, we provide various DLL APIs (implemented in C++ and Python), which can be divided into three classes: I/O operation, module invoking, and result extraction.

1. I/O Operation: Load texts or LTML files and convert them into DOM (Document Object Model); Save DOM to XML files.

2. Module Invoking: Invoke the 6 Chinese processing modules.

3. Result Extraction: Get the results produced by the modules.

Through invoking these APIs, users can accomplish some NLP tasks simply and conveniently. Assuming that we want to get the part-of-speech tags of a document, we can implement it with Python programming language easily as shown in Figure 2.



```
from ltp_interface import *
CreateDOMFromTxt("test.txt") # Load a text
POStag()                        # Invoke POS tagger
for i in range( CountSentenceInDocument() ):
      # Handle each sentence in a document
      word_list = GetWordsFromSentence(i)    # Get words
      pos_list = GetPOSsFromSentence(i)       #    Get POS
......
```

Figure 2: LTP Python API example

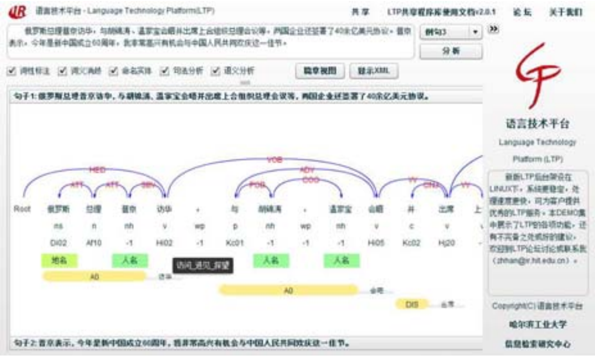
However, the DLL API has some shortcomings. First, it only can be used on Microsoft Windows machines. Second, users must download huge model files when LTP is updated. Third, LTP needs a high performance machine to run. All of above problems prevent from its widespread applications.

2.5 Web Service

In recent years, the Internet has become a platform where we can acquire all kinds of services. Users can build their own applications using LTP Web services conveniently. The LTP Web service has the following four advantages:

1. No need to setup LTP system.

2. No need to burden hardware to run LTP.



俄罗斯总理普京访华，与胡锦涛、温家宝会晤并出席上合组织总理会议等，两国企业还签署了40余亿美元协议。普京表示，今年是新中国成立60周年，我非常高兴有机会与中国人民共同欢庆这一佳节。

句子1: 俄罗斯总理普京访华，与胡锦涛、温家宝会晤并出席上合组织总理会议等，两国企业还签署了40余亿美元协议。

句子2: 普京表示，今年是新中国成立60周年，我非常高兴有机会与中国人民共同欢庆这一佳节。

最新LTP后台架设在LINUX下，系统更稳定，处理速度更快，可为客户提供优秀的LTP服务。本DEMO集中展示了LTP的各项功能，还有不完善之处或好的建议，欢迎到LTP论坛讨论或联系我(zhhan@ir.hit.edu.cn)。

Figure 3: Sentence processing result

3. Update promptly and smoothly.

4. Cross most operating systems and programming languages.

2.6 Visualization

A clear visualization can help researchers to examine processing results. We develop an cross-platform and cross-browser visualization tool with FLEX technology, which can be used easily without installing any excess software.

Figure 3 shows the integrated sentence processing results. The Rows 1 to 4 are the WordSeg, POSTag, WSD, and NER results. The last rows are the SRL results for different predicates. The syntactic dependency Parser tree is shown above with relation labels.

2.7 Sharing

We have been sharing LTP freely for academic purposes[4]. Until now, more than 350 worldwide research institutes have shared LTP with license. Some famous IT corporations of China, such as HuaWei[5] and Kingsoft[6], have bought LTP's commercial license. According to incompletely statistics, there are more than 60 publications which cited LTP, and the LTP web site has more than 30 unique visitors per day on the average.

3 Conclusion and Future Work

In this paper we describe an integrated Chinese processing platform, LTP. Based on XML data presentation, it provides a suite of high performance NLP modules invoked with DLL or Web service APIs, a visualization environment and a set of corpora.

Acknowledgement

This work was supported by National Natural Science Foundation of China (NSFC) via grant 60803093, 60975055, the "863" National High-Tech Research and Development of China via grant 2008AA01Z144, and Natural Scientific Research Innovation Foundation in Harbin Institute of Technology (HIT.NSRIF.2009069).

References

Berger, Adam L., Vincent J. Della Pietra, and Stephen A. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Comput. Linguist.*, 22(1):39–71.

Che, Wanxiang, Zhenghua Li, Yongqiang Li, Yuhang Guo, Bing Qin, and Ting Liu. 2009. Multilingual dependency-based syntactic and semantic parsing. In *CoNLL 2009*, pages 49–54, Boulder, Colorado, June.

Guo, Yuhang, Wanxiang Che, Yuxuan Hu, Wei Zhang, and Ting Liu. 2007. Hit-ir-wsd: A wsd system for english lexical sample task. In *SemEval-2007*, pages 165–168.

Hajič, Jan, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *CoNLL 2009*, pages 1–18, Boulder, Colorado, June.

Lafferty, John, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML 2001*, pages 282–289. Morgan Kaufmann, San Francisco, CA.

Liu, Ting, Jinshan Ma, and Sheng Li. 2006. Building a dependency treebank for improving Chinese parser. *Journal of Chinese Language and Computing*, 16(4):207–224.

Wang, Lijie, Wanxiang Che, and Ting Liu. 2009. An SVMTool-based Chinese POS Tagger. *Journal of Chinese Information Processing*, 23(4):16–22.

[4] http://ir.hit.edu.cn/demo/ltp/Sharing_Plan.htm

[5] http://www.huawei.com/

[6] http://www.kingsoft.com/