

Development of Part of Speech Tagging and Syntactic Analysis Software for Chinese Text

By

Daniel Tianhang Hu

Submitted to the Department of Electrical Engineering and Computer Science in Partial Fulfillment of the Requirements of the Degrees of

Bachelors of Science in Electrical Engineering and Computer Science
and
Masters of Engineering in Electrical Engineering and Computer Science

Abstract

The goal of this project is to complete initial development of Chinese Automatic Content Extraction software. This Chinese processing system is integrated with Alembic system (MITRE in-house NLP software for text processing), and has complete segmentation, part of speech tagging, and named entity detection capabilities.

First phase development involves establishing segmentation and part of speech tagging capabilities. Three different segmentation algorithms are implemented and integrated with the Alembic system, and one off-the-shelf software is tested and described. Different segmentation and part of speech guidelines are explored. Efforts include compiling different part of speech wordlists, segmentation wordlists, and supplementary wordlists to aid in segmentation, part of speech tagging, and machine training. Results are on par with industry standards. We found that replacement of the wordlists and resources within the existing segmentation/POS system can result in significant improvements. State-of-the-art software from external sources is shown to work well with the Alembic environment. Should we choose not to develop our own segmenter, license for OTS software can be readily obtained.

Second phase development involves named entity detection. Training is done with pre-tagged, cleaned, and re-formatted Chinese newswire data. While training shows promise, higher performances are quickly achieved through manual rule generation based on analysis of entity POS formations and additional specialized wordlists.

Throughout the development effort, guidelines are tailored to better meet the needs of Entity Detection and Tracking, and Relations Detection and Characterization.

Thesis Supervisor:

Dr. Amar Gupta

Co-Director PROFIT Initiative

Massachusetts Institute of Technology

Acknowledgement

I would like to thank Dr. Amar Gupta for providing this opportunity to work with MITRE Corporation. His support has been non-wavering for the past year. Dr. Doug Norman has extended to me a helping hand at the most critical of all moments and I am forever grateful to him for his believe in me.

I would also like to thank Dr. David Day and Dr. Mark Vilain for guidance from day one of this project. Their juggling act between numerous projects and meetings and hours with me over the computer is unbelievable. I also want to thank Dr. Qian Hu for helping me in every detail of the project and the thesis, and for being such a wonderful friend. I have to thank Ben Wellner for his wonderful tutorials, which has make impossible tasks simple. I want to extend my gratitude to all of the NLP Group. I have found a helping hand whenever I needed it.

My dear friends: I couldn't have survived through these five years if it was not for your understanding and support

Mom and dad: everything I have successfully accomplished in life can be accredited back to your absolute love and positive influence.

*The woods are lovely, dark and deep,
But I have promises to keep,
And miles to go before I sleep,
And miles to go before I sleep.*

-Robert Frost (1874-1963)

Contents

1	Organization of the Thesis	P6
2	Introduction	P7
	2.2 Building Chinese text processing system based on Alembic system ...	P8
	2.3 Three phases	P9
	2.4 Previous studies	P10
	2.5 Resource challenges	P13
3	Background	P15
	3.2 Alembic system	P15
	3.3 Chinese processing system	P18
	3.4 Segmentation problem in Chinese	P21
	3.5 Guideline for tagging documents	P25
	3.6 Part of speech tagging problem in Chinese	P26
	3.7 Phrasing problem	P29
	3.8 EDT and RDC	P33
	3.9 Resources: Treebank and PKU People's Daily data	P34
4	Methods	P37
	4.2 Sentence tagging	P37
	4.3 Segmentation	P38
	4.3.1 MMS	P38
	4.3.2 Personal Names	P42
	4.3.3 Heuristic Rule Based system	P44
	4.3.4 Unigram frequency model	P46
	4.3.5 PKU OTS software	P47
	4.3.6 Rule based Alembic segmentation module	P49
	4.3.7 LISP Part of Speech Tagging	P50
	4.4 Phrasing system in Alembic and OTC software	P52
	4.4.1 Training	P53
	4.4.2 Testing and scoring and error tracking	P54
	4.4.3 Compiling the frequency data	P57
	4.4.4 Generating wordlists to aid tagging sequences	P59

4.4.5	Manually Generated rules	P60
5	Results	P62
5.2	Results from phase 1	P62
5.2.1	Sentence tagging	P62
5.2.2	Segmentation	P62
5.2.3	Part of speech tagging	P63
5.3	Results from phase 2	P66
5.3.1	Machine Learned Rules for NE Task	P66
5.3.2	Improvements with Addition of Custom Wordlists and Manually Generated Rules for NE Task	P68
6	Discussion	P70
6.2	Separation of Phase 1 and Phase 2	P70
6.3	Segmenation	P70
6.4	Post-process for OTS module	P71
7	Conclusion	P72
8	Bibliography, references, and resources	P73
8.2	Web based resources	P73
8.3	Academic papers	P73
8.4	MITRE reports	P75
8.5	Operating Manuals and others	P75
Appendix A	P77
A1	Sample sentence tagged output	P77
A2	Sample segmentation output from MMS	P78
A3	Sample segmentation output from three word model	P79
A4	Sample segmentation output from Unigram model	P81
A5	Sample segmentation output from PKU segmenter	P82
Appendix B	P84
B1	Sample spec file	P84
B2	Sample learned rules file for phrasing	P84
B3	Manually Written Rules for Organization Tagging	P87
B4	MMS Segmenter	P89

B5 Translating wordlists from Web based dictionary	P95
Appendix C	P97
C1 Evaluation output of the trained phraser	P97
C2 Evaluation output of the manual phraser	P99
Appendix D	P100
D1 Surname list	P100
D2 Foreign phonetic translations characters	P101
D3 Organization and locations suffix	P102
Appendix E	P103
E1 University of Pennsylvania Chinese Treebank project	P103
E2 Beijing University Institute of Computational Linguistics	
People's Daily project	P104
E3 MET 2 and BBN data	P105
E4 Frequency data from PKU People's Daily project	P107
E5 Frequency data from U. of Penn. Treebank project	P108
E6 Sample organization part-of-speech frequency results from	
People's Daily	P109
E7 Phrase Tagged People's Daily Data (using Manually generated rules) ...	P109
E8 Correctly tagged People's Daily data	P110

1. Organization of Thesis

This thesis covers a broad range of topics. The introduction gives an overview of the system that we plan to develop. The background section sets up the problems that we face when building the system. This includes a more detailed description of the software system, the guidelines for the task that we are trying to perform, and the unique problems when dealing with Chinese. After the problems are presented, the methods section will provide the solutions we have envisioned. It will include information on how we utilize different resources, and different ways we can solve the same problem. The results section will present a summary of outputs from the various process schemes described in the method section. The discussion section following the results section will talk about some of the unresolved issues in our system. Finally, we will conclude with a very short summary of the discoveries.

2. Introduction

The accumulation of information in this electronic age is astonishing. Yet we have very little intelligent tools that will help individuals manage this goliath and help make critical decisions. Natural language processing research communities are looking closely at this problem and are attempting to build intelligent software that can “read” and “understand” as a person. This research is not only limited to English, however. As information in other languages multiplies at an even higher rate on the web, more attention has being placed on building a multi-lingual information-processing system.

One such language of interest is Chinese. Used by a fifth of the world’s population, Chinese language is playing an ever-expanding role on the World Wide Web for information exchange. Therefore, Chinese language processing is of great interest to government sponsors and people in the research community. Past efforts at processing Chinese have been undertaken by numerous research institutions and company. The primary focus of those researches has separately been segmentation, part of speech tagging, transliteration, and parallel corpus computing. Those researches delved deeply into the finite points of Chinese language, and addressed many important linguistic and computational problems. This project has a different goal. We are aimed at building a complete end to end system that is capable of doing Entity Detection and Tracking, and Relation Detection and Characterization, while utilizing the most advanced methods and results from other independent Chinese NLP efforts. The project is named ACEC, which stand for Automatic Content Extraction for Chinese.

In this introduction chapter, we hope to address three general topics. First we will introduce the Alembic processing environment. Then we will address the three phases of the project. Then we will briefly talk about the existing studies and language resources for Chinese language. Finally we will talk about our resource limitations.

2.2 Building Chinese Text Processing System Based on Alembic System

This end-to-end Chinese processing system will be based on an English processing system called the Alembic.

“Alembic software project for English arose out of the congruence of a need and of a technology. As information systems become increasingly ubiquitous in the worlds of work, government, and even the home, so grows the need for improving the acuity of information systems.

The principal value of Alembic system is that it provides fast and accurate models of language to identify aspects of a text that are of general interest to information system users. For instance, Alembic can identify proper names and classify them as to whether they designate people, places, organizations, and others. In addition, Alembic can be tailored to search for particular classes of events and to report them in a database format.”

1

The technical approach underlying Alembic is a new class of trainable language processing systems, rule sequence processors. These systems can either be configured to a new task by hand, or they can be configured automatically by providing a learning procedure with a collection of examples. In the latter case, the learning procedure automatically generalizes the examples into the same kind of linguistic model a human engineer would create by hand. This allows for unprecedented ease of development, since Alembic can be configured to an entirely new task by non-expert users who only need to create the collection of examples. This collection can be readily and rapidly authored with the Alembic Workbench².

¹ Description of the Alembic System Used for MUC-6, MITRE 97

² Summarized from <http://www.mitre.org/resources/centers/it/g063/alembic.html>

In addition to trainability, the rule sequence processors underlying Alembic are extremely fast, and confer on the system one of the highest processing throughput rates in the literature.³

The technical benefits of Alembic stated above can be adapted to a multilingual system. In fact, the rule based system and training capability make it easy to be adapted for other languages quickly and at a low cost. With minimum adjustment to the modules of the software system, and the additions of new modules, languages such as Spanish and Chinese can be added to the language capability.

The end goal is to have a Chinese text processing system that can achieve similar results as the English processing system. More details about Alembic processing system and how it is modified will be presented in the background and the methods section.

2.3 Three Phases of the Project

There are roughly three phases to the overall ACEC project. The first phase is exploration of text parsing and part-of-speech tagging on Chinese. The second phase is exploration of named entity tagging, and nominal entity phrasing. The third phase is Entity Detection and Tracking, and Relations Detection and Characterization. This thesis paper primarily focuses on first and second phase of this process.

Alembic system is a highly adaptable system of rule based learning machine. But to achieve the high quality results, a preprocessing module need to be added first to address the fundamental differences between Chinese and English. First, Chinese does not have overt word boundaries as English. Secondly, Chinese doesn't use inflection to mark verb tenses or part of speech. Thirdly, some Chinese expressions do not have direct counters in English, making direct mapping of English rules to Chinese impossible. These differences can be addressed through an initial module that performs several functions: 1, segmentation, and 2, parsing of sentences into meaningful word chunks, and 3,

³ Quoted from <http://www.mitre.org/resources/centers/it/g063/alembic.html>

importation of a new set of tags that could address difference in semantics. This initial module could then adapt Chinese text to a familiar English format and the rest of the processing may be done with more familiar methods. More details about the different segmentation and part-of-speech systems will be presented in the background and methods section.

The named entity tagging and nominal entity tagging performed on the second phase of the project will use much of the pre-constructed algorithms in Alembic language processing environment, namely, the training capabilities to do NE phrasing, and the flexible rules environment to construct manual syntactic rules. We will also be able to add wordlists and lexicon to this array of processing tools.

This thesis will not deal closely with EDT and RDC, which is the third phase of ACEC project. However, the design considerations for earlier systems are closely tied with requirements of EDT and RDC. Such design considerations are explicitly noted. The discussion section of this paper will also talk briefly about initial explorations of EDT and RDC, but we do not have results to back up the our theories or assertions.

Phase I Sentence parsing Segmentation Part of Speech tagging	Phase II Named Entity Tagging Lexicon resource generation Template building	Phase III* EDT and RDC Inference
------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------	-----------------------------------------------

Table 1 Phase I and Phase II are the focus of this thesis project. Phase III is explored but will not be part of the results

2.4 Previous Studies

Chinese is such a commonly used language in the world that prior study in this language is extensive. We have done extensive research into the areas that are relevant to this project. The sponsor's guideline explicitly encouraged the use of off-the-shelf (OTS) software whenever those are available and meet the performance specifications. Thus, a big part of this project is identifying such resources, evaluating the performance, and integrating acceptable outside resources to our system and putting together the separate

pieces that is the state of the art in their own respective fields. The most important criteria for evaluation and acceptance of the outside resources are the accuracy of segmentation and part of speech tagging, and the lexicon coverage. Named Entity tagging has not been a major focus of prior research for Chinese. Thus technology for NE task stem from the NE tagging capabilities of Alembic English processing system.

A wealth of information exists for tokenizing Chinese, chunking an otherwise contentious string of character with no white space in between into meaningful units of words. We have found good versions of segmentation software from University of New Mexico, University of Illinois, and some online sites⁴. These software include Erik Peterson's implementation of the Maximum Matching String Algorithm⁵, Chih-Hao Tsai's implementation of 3 word look ahead model⁶. There are also articles on statistical methods⁷, heuristic rule based methods⁸, Chinese name identification methods⁹, foreign name identification methods¹⁰, and segmentation methods that uses part of speech information and Viterbi algorithm¹¹.

The first phase of the project involves putting together the wealth of information already produce by other people to complete the segmentation and part of speech task. We are attempting to emulate some of those methods described by the research articles and adapted downloadable codes into our system. The codes that are directly used are segmentation codes. Some of the important methods we explored are maximum matching, three word look ahead, and unigram frequency model. Adapted into these codes are important Chinese and foreign name identification models that included

⁴ See reference for web based materials

⁵ <http://www.MandarinTools.com>

⁶ <http://www.geocities.com/hao510/mmseg/> is the location of the code. Algorithm lifted from 1992 paper by Chen and Liu.

⁷ Richard Sproat and Nancy Chang, 1994

⁸ Chen and Liu, 1992 COLING

⁹ Large Corpus Based Methods for Chinese personal Name Recognition, *Journal of Chinese Information Processing*, 6(3):7-15

¹⁰ Large Corpus Based Methods for Chinese personal Name Recognition, *Journal of Chinese Information Processing*, 6(3):7-15

¹¹ David D. Palmer and John D. Burger, Chinese Word Segmentation and Information Retrieval, 1997

valuable comprehensive wordlists on Chinese surnames and etc. These wordlists are build from past research efforts and improvements with the current pass.

Aside from the segmentation resources we collected or developed, there is also hand-annotated data critical to this whole project. As stressed earlier, one of the chief benefits of Alembic system is it's learning capabilities. Machine training requires large amount of manually corrected data. Due to limited time and human resources, manual tagging our own data is impossible. The next best thing is to use data from outside resources. Some of the data that are used in this project come from University of Pennsylvania Treebank project¹² and Beijing University ICL's People's Daily data¹³. Their segmentation guidelines are not quite the same. Nor are their part of speech tag sets. More details about these two corpuses and some other segmented data sets will be provided in the backgrounds section. We have also attached sample files from these important corpus in the Appendix, and tables that provides definition to different tag sets¹⁴.

MITRE also previously studied processing Chinese language¹⁵. This was done without the support of staff with Chinese language background. Thus, all the performance levels were achieved by machine learning. There were resources that were developed in conjunction with the previous effort. Those include derived wordlists of national names, geographical locations, foreign names, and world capitals¹⁶. Those wordlists are of limited use, because editing was not possible due to lack of personnel with Chinese knowledge. During this second initiative at processing Chinese, those resources are cleaned up and reorganized. In additions, English wordlists were used to build a parallel corpus in Chinese with web mining software developed for this project.

There are large numbers of unnamed resources for this project. They are mentioned either in the thesis or listed under bibliography information. But even with careful accounting, there is going to be missing pieces. We will attempt to do a through job in

¹² Appendix E1

¹³ Appendix E2

¹⁴ Table 4

¹⁵ David D. Palmer and John D. Burger, Chinese Word Segmentation and Information Retrieval, 1997

¹⁶ Figure 11

explaining the most important parts to our system and how each of those parts plays an irreplaceable role. However, we apologize in advance for any thing missing or misrepresented.

2.5 Resource Challenges

Finally, we would like to highlight some of the limitation of this project before we finish introduction and go into further detail. The first limitation is time. This project was initiated on February 2001. The time that the thesis is being written is May 2001. In three month, we completed initial exploration into phase I and phase II of the end-to-end system, and is delving into phase III. The results are good, but are still far from the numbers of the parallel English system. Better results are almost guaranteed with better wordlists, more training, and more detailed rules. But the necessary hours needed for those activities just do not exist.

Secondly, funding for this project has not being 100%. Some of the obviously superior resources have not materialized due to stalls in the licensing department. We can only complete analysis on parts where we can generate output, and fake the input into the later modules that requires the missing link. This is unfortunate because the numbers are bond to change when real modules come to replace the imaginary one. We will take that into account when writing the results and provide further projections in the discussion section.

Lastly, we are working with limited linguistic resources. In a perfect world, we will hand generate training data with the best part-of-speech and segmentation. This will optimize the capabilities of the named entity detection rules and provide the best foundation for future EDT and RDC. But this is a time consuming and frustrating process. Thus, a compromise is reached by using less-than-perfect resources from other institutions and piecing them together in a patch-work of training data and derived wordlists.

Even with the limitations mentioned above, our exploration for the first and second phase of building a Chinese text processing system has been thorough and detailed. This report will be useful to others carrying out the rest of ACE Chinese project, both as a reference to all the available resources and as a general guideline.

3. Background

The background section defines the problems we are facing when building a Chinese text processing system. The first section of the background will describe in more detail the Alembic system mentioned in the introduction. The second section will follow with proposed design for Chinese processing systems. The third section will define the Chinese segmentation problem in more detail and highlight the different guidelines towards this problem. The fourth section talks about the tagging structure used. The fifth section describes the part-of-speech tagging problem and ambiguities. We will also provide some examples of how Chinese is different from English. An important table of different part of speech tag sets will be provided. Closely related to part-of-speech tagging is the phrasing problem. This is described in the fifth section. In this large section we will also provide guidelines to named entity tagging. The sixth section provides guidelines to EDT and RDC as defined by ACE groups. Here we will also detail some of the processing steps needed to do EDT and RDC in the English system. In the seventh section more will be said about two of the important annotated corpuses: Penn treebank, and PKU People's Daily.

3.2 Alembic Processing System¹⁷

The Alembic system divides roughly into three general sections, a C and UNIX based pre-processor, also called the pre-alembic, a LISP-based phraser and syntactic analyzer, and a LISP based inference system. But as we shall see, the boundaries are not as clear once we enter the domain of LISP system, as the flow and accumulation of data is done from all levels and it can not be modeled as a straight series of one input, one output modules.

For the English language, the pre-processor, also called by the name pre-alembic, is a patch work of code that dates back to the earliest text processing efforts at MITRE. The preprocessor divides into two units. First step of pre-alembic is called zoning. Often the

¹⁷ Description of the Alembic System Used for MUC-6, 1997

input into language processing software is not clean, coming from various different sources, with various different tags and errors. Zoning is the process which strips the document of symbols that might cause errors. Then it will pass the document onto the sentence/lexicon detection module. This module will identify sentence boundaries, and place <LEX></LEX> tags around words and end-of-sentence punctuations. The LEX tags would include code for part of speech information about the words. Such as “noun” or “verb” or “adjective”. The output of the pre-alembic system is a clean, marked up piece of document, which is ready to be further processed by the LISP system.

The building block for LISP processing environment are lists of rules. Each rule implicitly contains a simple linguistic property. For example, we might have a rule that equivocally says: “*if a noun follows an adjective, that adjective goes to modifying this noun.*” This rule might apply 90% of the times. For the 10% of the exception, other simple, straightforward rules will attempt to patch up the errors. The list of rules for each module is applied serially to perform the tagging required of that list, and every rule is applied only once. For a complete phrasing module, there will typically be one to two hundred rules. The tagging specified by the most recently applied rules overwrites whatever tagging has been done before. Thus the general trend in writing rules is to go from broad to narrow. The benefits of this rule-based system include speed, flexibility, and machine learning capabilities. Because each rule is almost like a simple observation, a computer with a large training data can generate and re-arrange the rules until it is able to get very high scores on the training corpus.

The modules for the LISP phraser start with the name entity tagger. This system is responsible for tagging all personal names, locations, organizations, geo-social-political groups (GSP), and governments. Then the tagged data is passed to the phraser module, which is responsible for expanding the nominal entities into nominal phrases. For example, the following phrase, “*Senator Kerry of Massachusetts*”, would first be tagged with “<post>*Senator*</post> <name>*Kerry*</name> of <location>*Massachusetts*</location>” in the named entity tagger, this sequence would be expanded into “<person><post>*Senator*</post> <name>*Kerry*</name> of <location>*Massachusetts*

Figure 1 The complete schematics of the Alembic English processing system.

</location></person>” in the phraser module. More details about phrasing will be provide in later section of this chapter.

The tagged data would then be passed on to the grammatical relations module. Like the name implies, this module is responsible for establishing the relationship between the nominal and verbal phrases.

The final part of the LISP processing system is the co-reference module. After the grammatical relationships have been established in each sentence, this module will attempt to connect the nominal phrases.

All of the main LISP modules feed into a temporary data bank similar to a template. As more information is gathered, more tokens and ties are added into the data lists. Higher level processing such as Entity Detection and Tracking, and summarization will use this database to complete the job.

3.3 Chinese Processing System

A different schematic than the one for English system is needed for processing Chinese. The problems unique to Chinese that will not be resolved using existing framework is segmentation and part-of-speech tagging. More details about the problem of segmentation and part-of-speech tagging will be provided in sections 3.4 and 3.5. In this section, we will talk about some possible system solutions, and introduce the resources that we are exploring. To summarize our needs, a component needs to be created that will take the input from the zoning module and output segmented, part-of-speech tagged, and sentence tagged text. This data will then be passed on to the LISP phraser and inference module built into Alembic, and the rest is similar to the English system. This data flow illustrated in figure 2 and in the example below.

Input:

(Our nation is currently moving from subsistence farming to commercial farming)

Output: <S><t> </t><n> </n><n> </n><d> </d><ppos> </ppos><vn>

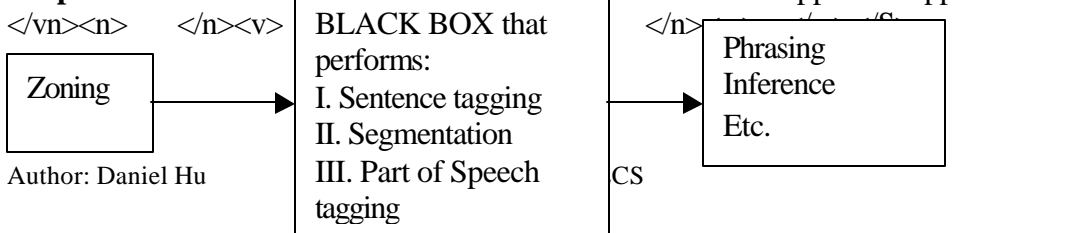


Figure 2

In this input/output box, we can treat segmentation and part-of-speech tagging separately, or we can treat them together. The argument for treating them separately is increased processing speed and ease of development. On the other hand, part-of-speech information may aid in clarifying segmentation ambiguities. The desired solution would be to build and test both systems. But there are resource limitations. To build a combined system from scratch, we needed a very good part-of-speech dictionary, which is unavailable. So by default, the system developed and tested in-house all treated part-of-speech and segmentation separately. The first stage is the segmentation. Then the output from the segmenter is fed into a LISP based part-of-speech tagger using a very limited 50,000 word part-of-speech lexicon.

Segmentation of Chinese sentences is always treated one of two ways. There is the purely statistical approach, and then there is the lexicon approach¹⁸. Past researches have generally shown that lexical approach is more effective and requires less processing power. Thus, all segmentation paths that we are exploring fit under the lexical approach category. Chinese is read from left to right, and majority of the segments can be determined by a technique called maximum matching string (MMS)¹⁹. Existing code for this method can be downloaded from www.MandarinTools.com, developed by Peter Erikson. This is our baseline. Other techniques that go beyond this simple algorithm are maximum matching string with 3 word look ahead (MMS3) developed by Tsai²⁰ based on Chen and Liu's 1992 paper²¹, and 3 word look ahead model that take advantage of frequency data that is modified from the MMS model. The following figure illustrates the data flow diagram.

¹⁸ Richard Sproat and Nancy Chang, 1994

¹⁹ <http://www.MandarinTools.com>

²⁰ <http://www.geocities.com/ha0510/mmseg/>

²¹ Chen and Liu, 1992

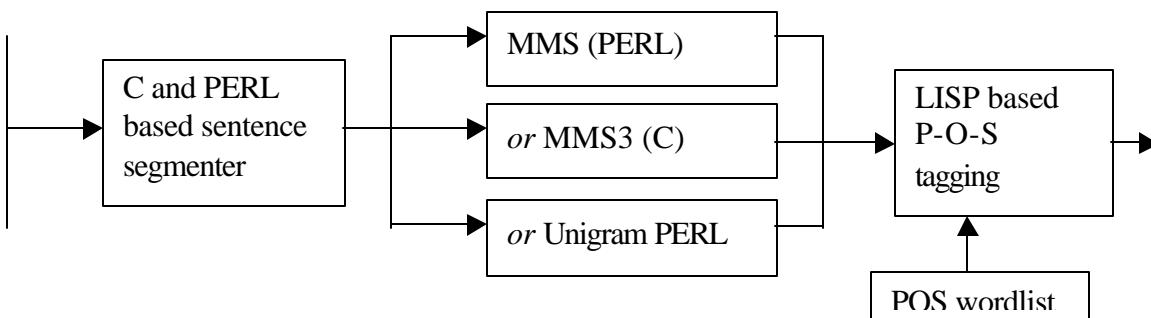


Figure 3 The three different modules that can do segmentation combined with a LISP based part-of-speech tagging module.

The alternative to treating segmentation and part-of-speech tagging apart is to build them into a single module. This method is effectively exploited by an off-the-shelf (OTS) software from Beijing University Institute of Computational Linguistics (PKU ICL)²². The output of this system is not in the format that we can use in the Alembic processing environment. So post-processing tools are added to the data path to further expand the tags and create the desired data format.

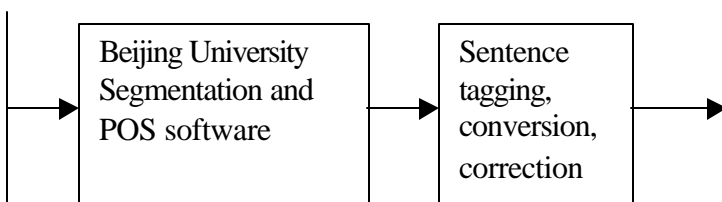


Figure 4 Installing Beijing University segmentation and POS software into Alembic.

With the correct segmentation and part-of-speech information, the data is then fed into the Alembic rule based modules. The first pass is the named entities tagger just like the English system. Then it is passed on to the phraser, which expands the named entity tags into nominal phrases. All the while information is fed into the template. The exploration in this project is limited to finding the optimal solution to segmentation and part-of-speech tagging, and named entity task.

²² <http://www.icl.pku.edu.cn/nlp-tools/segtagtest.htm>

3.4 Segmentation Problem in Chinese

Predicating the understanding of processing techniques for Chinese, we first need to establish the basics of Chinese grammar and word. While Chinese have a rich tradition of phonology, grammar studies are fairly recent on the historical scale. The original studies on the Chinese grammar were done as the result of Western language influences. The concept of a word is intuitive to the native speaker, but an effort to establish a basic guideline for singling out a word and giving it a grammatical role was not undertaken until late 19th century after the introduction of Western linguistic concepts. So grammatical concept in Chinese is not completely alien to the grammatical concept of romantic languages. Yet, there are great many instances of ambiguity when we attempt to dissect Chinese language using the basic linguistic rules provided for English. Thus arose the need to import a different set of guidelines for handling Chinese.

The first difficulty in analyzing Chinese is segmentation and definition of a word. The most basic unit of the Chinese language is a morpheme²³. A morpheme is one Chinese character. On a computer, Chinese morpheme can be expressed in GB and BIG5 encoding, where each character is made up of 8 bytes. In Unicode, this character may be encoded in either 16 or 32 bytes depending on which version of Unicode. There are bounded morphemes and free morphemes. Free morphemes are characters that retain an independent meaning and can appear on their own. Bounded morphemes are those that do not appear independently. An example of a free morpheme is (gou – dog). While this character often appear on it's own, it can also enter into polysyllable combination with other morphemes to form words, such as (mugou – bitch), or (guoxiong – bear). Absolute bounded morphemes are much more rare in Chinese. An example of the bounded morpheme is (jili – caltrop, a type of plant), where the first character (ji) has no meaning independent from the second character. A sentence in Chinese is formed of a string of morphemes without space to separate them. The reader supplies intuitive boundary to the words and construct them into logical sequences. Classical Chinese literature would mainly use one morpheme unit as the basic word, such as the writings of

²³ Jerry Norman, Chinese, 1998

traditional courtiers of imperial China. But preceding the republic era, Chinese scholars and writers began to adopt *bahua*, or a written form of the spoken language. *Bahua* contained many more polysyllable words. Slight variation on this historical language development survive today as the official language of China, called Mandarin. Mandarin possess many polysyllable words, yet, the written form of Mandarin follow the tradition of no space boundary between words²⁴. This poses a special problem to the linguistic computation, which bases much of the algorithm on the fundamental unit of a word.

The most encompassing law for segmentation is “*strings of characters which are the smallest unit of meaningful sequences that plays the same role in or out of context*”. We will attempt to apply this rule whenever we can towards the problems of segmentation of Chinese. But there are many cases where this rules is applied with a lot of ambiguity. For instance, the sentence

“ ”

(zhong guo lian xiang gong si de fu cong jing li fang wen pu dong te qu – China
Liangxiang company’s deputy general manager visits Pudong special economic region)

contains (fu cong jing li – deputy general manager) and (special economic region). If we apply the strictest sense of the *smallest unit of meaning* rule, we will get _ _ (deputy_general_manager) and _ _ (special_region). Yet, most linguist would agree with the first case of segmentation, but would disagree with the second case of segmentation. This is because of a second law. Which states that “*bound morphemes together when, if treated independently, their meaning are highly adaptable*”. In this example, are treated as one word even though they are both free morphemes because the is almost always presided by some kind modifier and easily form polysyllable words. In another word, both morphemes are very versatile²⁵. But this introduces a lot of confusion because the versatility of a morpheme is a highly subjective topic, and different

²⁴ Jerry Norman, Chinese, 1998

²⁵ Jerry Norman, Chinese, 1998

linguist would have different ideas of how to segment the same sentence. Often, two annotators will have 10% disagreement over the same corpus.

For the purpose of our project, we have not adopted any specific guideline. The rule of thumb will still be “*smallest unit of meaning*”. In any case, the specifics of segmentation have more to do with easy of information extraction in the down stream processing modules. The word unit has to be small enough such that 1. accurate part-of-speech information can be assigned and 2. we will not have to take words apart in later modules to do EDT and RDC. An example would be comparing (fuxiaozhang – vice chancellor) with (xiaozhang – chancellor). We most definitely would want (vice chancellor) displayed as _ (vice_chancellor) because it would be easier to establish relationship between vice chancellor and chancellor in downstream processing, instead of treating them as completely independent positions.

The next immediate problem after determining what constitutes a word segment is how to actually extract this information out of a sentence. A native reader of Chinese will be able to determine the correct segmentation meaning from context.

The corpus that we will be primarily working with will be official Chinese newswires and certain legal documents. These texts will be written in *putonghua*, or “common language”. Minus some special articles and summary reports, the language used in these articles would follow journalistic tone, giving us ample opportunity to come up with common rules. In addition, the polysyllable words will be easily found in the dictionary. Two sources of the journalistic text are explored. One is the Treebank project annotation of Xinhua news agency articles, and the other one is PKU ICL annotation of People’s Daily articles²⁶. Frequency of word length is presented in the following table.

	Xinhua News Agency		People’s Daily	
Total words	82,148		907,017	
1 morpheme	27,187	33.1%	337,724	37.2%
2 morpheme	47,216	57.5%	492,879	54.3%

²⁶ See Appendix E1, E2

3 morpheme	6,330	7.7%	51,116	5.6%
4 morpheme	857	1.0%	19,375	2.1%
5 morpheme	468	0.6%	5059	0.6%

Table 2 Word length distribution of Xinhua News Agency and People's Daily. Comparison of segmentation results from U. of Penn. Treebank project and PKU ICL annotation.

3.5 Guideline for Tagging Documents

Before introducing the part of speech tagging and phrasing problems, we will introduce the SGML tagging guidelines. For start, every article needs to have a correct set of HTML tags. Every processing module looks for <DOC> and <TXT> tags to begin processing. Instead of putting everything into words, a table is provided below to illustrate all the tagging needs.

Module output	Tags	Example
Newswire data	(dirty) HTML <i>missing or miss-labeled tags</i>	<DOC> <TEXT> <P> </P> </DOC>
Zoning	(clean) HTML	<DOC> <TEXT> <P> </P> </TEXT> </DOC>
Sentence Segmentation	<S></S>	<DOC> <TEXT> <P><S> </S></P> </TEXT> </DOC>
Segmentation and P.O.S. tagging	<LEX POS="v"> </LEX>	<DOC> <TEXT> <P><S><LEX POS="nr"> </LEX><LEX POS="v"> </LEX><LEX POS="nt"> </LEX><LEX POS="w"> </LEX></S></P> </TEXT> </DOC>
Named Entity Task	<ENAMEX TYPE ="ORG"></ENAMEX>	<DOC> <TEXT> <P><S><ENAMEX TYPE="PERSON"><LEX POS="nr"> </LEX></ENAMEX><LEX POS="v"> </LEX><ENAMEX TYPE="LOCATION"><LEX POS="nt"> </LEX></ENAMEX><LEX POS="w"> </LEX></S></P> </TEXT> </DOC>

Table 3

3.6 Part of Speech Tagging Problem for Chinese

Grammar is a direct result of the identification of a word. Sequences of words placed together in coherent and consistent way must be linked together through the concept of part-of-speech, because each word has a role of gluing the other words together or taking an active part in the meaning of the sentence. This concept is as Western to the Chinese language as is the concept of a word. But it has also been adapted to fit under the framework of Chinese language. In today's linguistic research community the major word classes (part of speech) enjoy fairly high degree of agreement. The scheme outlined below is close to that found in several widely used grammars.

There are six full word categories: nouns, verbs, adjectives, numerals, measures, and pronouns.²⁷ A noun is something that can be directly counted with a numeral: (sanliangche - 3 cars). A verb is a word that can be negated by (bu - not): (buqu - not going). Adjectives are generally considered to be a relative to the verb. They can also be negated by (bu - not) and can function independently as predicates. Numerals include classes of Chinese numbers and estimations of quantities, in addition to roman numerals. Measures are words that indicate unit for each noun. In the previous noun example, (sanliangche - three cars) is actually formed of (san-three), (liang - measurement for vehicles), and (che - car). Measures have similar function as “sheets” in “three *sheets* of paper” in English, but they are much more prevalent. Pronouns are substitute for a person or things. Syntactically they generally behave like nouns, but they do not admit modification. The third person pronoun is gender specific in certain writing, and unspecific in others. They are (ta - he, she, it). Other pronouns such as demonstrative pronouns include (zhe - this) and (na - that). Other words that commonly included with the pronouns are (shengme - what), (self), and (every).

²⁷ Jerry Norman, Chinese, 1998

The functional word classes are adverbs, prepositions, conjunctions, particles, interjections, and onomatopoeia.²⁸ Adverbs modify verbs and adjectives, and are usually bounded. Chinese prepositions originate from verbs, and are used to show various grammatical relationships. Conjunctions form different kind of connections: (he - and), (erqie - moreover), (yinwei - because), (suoyi - therefore). Particles are generally monosyllabic words used to show different grammatical relationships or subjective and modal overtones: (de), (ba). Interjections functions as asyntactic forms used to express warnings. Phonologically they contain irregularities such as elements that are not part of the regular phonemic inventory. Onomatopoeia are words which mimic the sounds of things. Syntactically they behave much like adverbs or nouns.

As in English, there are considerable overlaps between the different classes. One word or morpheme may belong in several categories. An example is (piping - criticize), which in this case is a verb. But we can use the same as a noun which means criticism. Below are two sentences that uses the same word with different part-of-speech:

(We should criticize this kind of behavior)

(Your criticism is justified)

This kind of overlap is a lot more prominent in classical Chinese literature. But in today's written language, it is no more prevalent than that of English. Thinking back towards the segmentation and part of speech tagging diagrams, we will have to build post processing modules for any imported software to make the necessary corrections. In the example above, we can write a rule that says that any segment tagged as verb would have the tag changed to noun if it came after the character (de).

This is just a brief summary of all the possible part-of-speech designations in Chinese. There are almost always subdivisions and new categories created when we deal with different linguistic camps. Two main groups we have looked at are the University of Pennsylvania's Chinese studies department and Beijing University's ICL. University

²⁸ Jerry Norman, Chinese, 1998

of Pennsylvania developed their part of speech tag set based on the requirements of the Chinese treebank project, an effort to completely parse Chinese sentences. PKU ICL developed their part-of-speech tag set mainly to aid computational linguistics. The tags from Beijing University are presented below. These are more important to our system because majority of the training was done based on corpus with these POS tags.

PKU POS	Definition
na	Noun used as adjective
Mg	Numeral used as a noun
vd	Verb used as adverb
nr	Personal Name
a	Adjective
ns	Nominal location
nt	Nominal organization
b	Negative
c	Conjunction
d	Adverb
vn	Verb that can also be a noun
e	Another form of interjection
ad	Adverbial adjective
f	Location
h	Head
i	Idiom
l	Transient word
n	Noun
an	Adjective that can also be a noun
o	Onomatopoeia
p	Prepositional
q	Quantity measure
r	Pronoun
s	Space
t	Time
u	Auxiliary
v	Verb
y	Tone

Figure 5 A partial list of PKU POS tags.

3.7 Phrasing Problem²⁹

Traditionally the phrasing problem for Chinese is viewed as two separate tasks. The first pass is the Named Entity Detection task as defined under the Multilingual Entity Task guidelines. The second pass is Entity Detection as defined under the EDT and RDC guidelines. The difference between those two tasks is coverage. The NE task has a more narrow focus, aimed at identifying unambiguous entities in any corpus while ignoring unspecific mentions. EDT and RDC task guideline has a broad focus. The end goal of EDT and RDC entity task is to find specific mentions (as specified in the NE task), and expand their association to unspecific mentions. So the EDT and RDC phrasing

²⁹ EDT Metonymy Annotation Guidelines Version 2.1 (2001), Multilingual Entity Task (MET) Definition for Chinese (1996)

guideline would cover specific mentions as well as some reference mentions. There are two different scoring processes for the NE task and EDT. They both take machine generated phrase tags and compare it to hand annotated phrase tags, and calculate a percentage called the f-measure (more about f-measure will be presented in the methods section, 4.5). The hand annotated phrases need to abide by the different task guidelines depending on which task we are trying to accomplish.

Under the sponsor's contract, our end goal is EDT and RDC. Traditional NE task is of less importance to the whole process. So our first phrasing pass is a hybrid of the traditional NE task and the complete EDT named entity detection task. In this design, a more inclusive second pass will make corrections on the first pass, and complete the EDT named entity phrasing task. This hybrid NE task is similar to the traditional NE task in that it only allows single NE tag around any phrase. The markup is also based on the <sgml> tag set used for NE. Traditional NE task can be broken down to three subcategories. These subtasks are responsible for capturing entity names, temporal expressions, and numeral expression, respectively. Entity names are mentions of organization, persons, and locations. We will present some of the entity tagging in more detail in the following chapters. But first we will take a closer look at the simpler task of marking up temporal and numeral expressions.

Temporal expressions are for “absolute” and “relative” time expressions³⁰. In the 6th Message Understanding Conference, the NE tagging guidelines for English only specified the tagging of absolute time expressions. This standard is relaxed and expanded for Chinese to include relative time expressions³¹. Absolute time expressions include the any specific mentions of time, date, and duration that are independent from the document. Relative time expressions include all other specific mentions of time such as “yesterday”, “last month”, “in a few hours”. Just like English, time expressions in Chinese have exclusive units that can be anchored to achieve high performance relatively easily³². The tagging process will be presented briefly in the methods section. But it will not be part of

³⁰ Multilingual Entity Task (MET) Definition for Chinese, 1996

³¹ Multilingual Entity Task (MET) Definition for Chinese, 1996

³² <http://www.MandarinTools.com>

the result because detailed analysis for the relatively simple task is not needed. Numeral expressions are mentions of any monetary amount or percentages. Similar to the temporal expression task, there are clear anchors to be used when dealing with these phrases. Rules which are used to tag these expressions will be briefly dealt with in the methods sections.

The more difficult task is the entity task. The uniqueness of Chinese means that much of the concepts which are used to handle the English entity detection task can not be applied. Chinese entities types have to be examined independently from their English counter parts. The three subcategories for entity in Chinese are persons, locations, and organizations. The hybrid technique that we have adopted for the first phrasing pass is also responsible for tagging some of a fourth category, namely Geo-Political-Entity (referred to as GPE from now). We will examine the problems with each of these categories separately.

As reader knows, Chinese do not have “capitalized” characters, so personal names can not be picked out like their English counterparts. On top of that, this is also a problem for segmentation because Chinese sentences contain no spatial separation. Political entity and locations might be resolved on the segmentation level with a good inclusive list, but personal names are a completely different story. There is an unlimited number of personal names possible, and these include all the Chinese names as well as translations of English names. So the segmentation of Chinese and English personal names have to be solved at the segmentation level. There are rules to forming a Chinese name, though, and a detailed description of how these rules are exploited is provided in the method section. To give a light overview: the method takes advantage of the fact that Chinese surnames come from a finite list, and given name is usually no more than two characters long.

This method does not work for foreign names. Fortunately, foreign names are usually sound translations that derive their phonetics from a finite set of Chinese characters. So we have methods that work with this characteristic to tag foreign names. In summary, a

majority of personal names will be resolved at segmentation level before the LISP system. The tags assigned to names at the segmentation level will be LEX tags with personal name designated “part-of-speech” information. Based on the “part-of-speech” information, phrase tags will be added around the personal names. At this stage of tagging, titles will not be included as part of the person, nor will other entity modifiers.

Phrasing of locations and organization is done in the LISP based rule modules. Unlike personal names, word components to locations and organization are already segmented when they have reached the phraser. The phraser will need to identify all the components that make up the organization or location and put them together into a phrase.

The general rule of thumb is to put together things that are not separated by possessive particles. (meiguoguoofangbu - American defense department) should be treated as organization, where as _ _ (meiguo_de_goufangbu - America’s defense department) would tagged separately as location and organization³³. Or if a superior organization or location precedes the immediate mention of a subordinate or representative organization, and they are not separated by a possessive, they should be tagged as one³⁴. An example of that would be (Chinese consulate stationed in US): Chinese government () is separated from consulate by the verb (zhu - stationed), but since this is not a possessive verb, the primary organization is combined with consulate to form a single organization. Similar rules apply for location. (zhongguobeijing - Beijing, China) is tagged as one location because they are not separated by a possessive.

Generic mentions are in most cases not tagged. Generic mentions constitute mentions of organization and location that is not identifiable out of context. Examples of such mentions are (fandudang - opposition party) and (bangongshi - office). The MET2 guidelines for NE tagging in Chinese specified that *US government* is treated

³³ Multilingual Entity Task (MET) Definition for Chinese, 1996

³⁴ Multilingual Entity Task (MET) Definition for Chinese, 1996

as a generic mention and not tagged. This rule is relaxed when we built our hybrid NE system. Any location linked with a government office is tagged as an organization. The MET2 guidelines further specified that each word may only have one NE tags. This rule is adhered to. Our tag set includes only ORGANIZATION, LOCATION, and PERSON. Any word will only be wrapped with one of these tags.

Under EDT guidelines, locations that make reference to an Organization or broad organization mentions are tagged as a GPE. *US Government* is one example of a GPE. Another example of GPE is (baigong - Whitehouse). In our NE task solution, we handled part of the GPE tagging by tagging as a location, and tagging *US Government* as an organization. This is far from complete, however, and much more needs to be done before EDT entity detection task is satisfied.

3.8 EDT and RDC

We will now talk briefly about the EDT and RDC task for the benefit of discussion. There are no results to show for EDT and RDC, but it is important to understand the final goal of the project in order to appreciate the intermediate steps. EDT and RDC stand for Entity Detection and Tracking, and Relations Detection and Characterization. The goal of EDT is to identify a specific entity and track its activities or characteristic through documents. The EDT task is actually a complex of different tasks.³⁵

1. The detection of entities. This is the bases on which all other tasks are built. This is for the detection of specific entities. The named entity phrasing module completes part of the task, although another processing layer need to be added to clarify between location, organization, and GPE.
2. The recognition of entity attributes. This is partly integrated with the first part. Distinctions need to be made as to the type of named entity. The tag set expand slightly as more specific labels are assigned.

³⁵ EDT Metonymy Annotation Guidelines Version 2.1, 2001

3. The detection of entity mentions. This measures the system's ability to associate a set of mentions with a specific underlying entity. This includes pronominal mentions.
4. The recognition of mention extent.³⁶

The RDC uses information from EDT and advance it one step further by finding relationship between the different entities that are supported by explicit or implicit textual reference. The target relations will be Role, AT, and Part, respectively linking person with organization, organization or person with location, and person or organization with person or organization. All three of the relations will contain fields for temporal attributes. In addition, Role relation will also carry Role attribute to classify the connection between two entities.³⁷

To satisfy EDT and RDC requirements, phrasing capability would have to be expanded to include title, pronominal mentions, some generic mentions, and all GPE. To this point, we have completed only part of the GPE tagging beyond the traditional NE task. But we believe that techniques developed for NE task may be easily adapted to building more sophisticated rules to tag the other mentions.

3.9 Treebank and PKU Data

University of Pennsylvania and Beijing University (PKU) have independently hand-annotated data available as a resource to computational linguists. This is a valuable tool for machine training over wide range of issues. University of Pennsylvania's effort was aimed at creating a 100,000 word corpus of Mandarin Chinese text with syntactic bracketing. Their effort consisted of two stages: the first stage is word segmentation and part-of-speech tagging and the second stage is syntactic bracketing. Beijing University's annotation effort was focused only on segmentation and part-of-speech tagging, although later efforts had basic phrase identification. The size of their corpus is much larger, at

³⁶ EDT Metonymy Annotation Guidelines Version 2.1, 2001

³⁷ Preliminary Guidelines for Relation Detection and Characterization (RDC), Draft Version 2.2, 2001

around 1 million words. Both research facilities used Chinese News data. University of Penn annotated Xinhua News Agency articles. PKU annotated one month of People's Daily. The tone of both source is very similar. For our purposes, segmentation and part-of-speech tagging was enough. The so the University of Penn data was stripped of the syntactic bracketing before being used for POS training and evaluation. The segmentation guidelines for both annotation efforts are similar. But there are small differences, and this is reflected in the percentage of words of different length (see table 2). Part of speech tagging is also different. Since University of Pennsylvania extended their effort to include syntactic analysis, their part of speech had less detail, and relied more on syntactic labels. PKU relied exclusively on part of speech and some limited nominal differentiation to carry all of the information, therefore they developed a larger tag set. Table 4 presents their POS tags in detail. The following is some sample data. Larger samples can be viewed in Appendix E.

University of Pennsylvania Treebank Data:

```
<S>
( (IP (NP-SBJ (NP-PN (NR   ))
      (NP (NN   )
        (NN   )
        (NN   )))
  (VP (VP (VV   )
    (AS   )
    (NP-OBJ (NP-APP (NN   )
                  (PU   )
                  (NN   )
                  (PU   )
                  (NN   )))
      (QP (CD   )
        (CLP (M   )))
        (NP (NN   ))))
    (PU   )
    (VP (NP-TMP (NT   ))
      (ADVP (AD   ))
      (VP (VV   ))))
    (PU   )))
</S>
```

Input into Alembic system would appear as:

```
<S><LEX POS="NR"> </LEX><LEX POS="NN"> </LEX><LEX
POS="NN"> </LEX><LEX POS="NN"> </LEX><LEX POS="VV"> </LEX><LEX
POS="AS"> </LEX><LEX POS="NN"> </LEX><LEX POS="PU"> </LEX><LEX
```

```

POS="NN"> </LEX><LEX POS="PU"> </LEX><LEX POS="NN"> </LEX><LEX
POS="CD"> </LEX><LEX POS="M"> </LEX><LEX POS="NN"> </LEX><LEX
POS="PU"> </LEX><LEX POS="NT"> </LEX><LEX POS="AD"> </LEX><LEX
POS="VV"> </LEX><LEX POS="PU"> </LEX></S>

```

Beijing University People's Daily Data:

```

19980110-01-001-023/m /v /v /v /n /vn /w /vn /w /n /w

```

Input into Alembic system would appear as:

```

<S><LEX POS="v"> </LEX><LEX POS="v"> </LEX><LEX POS="v"> </LEX><LEX
POS="n"> </LEX><LEX POS="vn"> </LEX><LEX POS="w"> </LEX><LEX
POS="vn"> </LEX><LEX POS="w"> </LEX><LEX POS="n"> </LEX><LEX
POS="w"> </LEX></S>

```

4. Method

The methods section will attempt to provide reader with details on how we developed various solutions to the problem and questions framed in the background section. We will be using tools developed by author of this paper or adaptations of resources available from other research labs. The organization of the methods section roughly corresponds to the flow of data through the system. The first section deals with sentence tagging and paragraph tagging, a prerequisite for anything passing through to the Alembic processing environment. Then we describe our segmentation efforts, going into detail about how we integrated three different segmenters into the system, along with a description of the OTS software from Beijing University. Imbedded in the segmentation problem is also the problem of name identification. The segmentation section will end with a conclusion and summary of pros and cons of the different methods we have examined. The next section provides solution to part-of-speech tagging. Through out the segmentation and POS section, we mention using various derived wordlists. The nest section will give details about how those wordlists are derived and merged. There is also an inventory of extra wordlists that will help the NE and EDT tasks. The next large step in this process, after Chinese text have been segmented and POS tagged, is NE tagging. We divided up the phrasing task into machine training, scoring, generating frequency data, manual wordlist development, and finally manual rule writing. Our efforts up till this point in time are satisfactorily described by the above summary.

4.2 Sentence tagging

Sentence tagging is a relatively simple task for Chinese. In English, we run into problems with ambiguous punctuations such as *Mr.* or *P.O.W.* In Chinese such mid-of-the-sentence punctuation would never use terminal punctuations such as the Chinese period () or the exclamation mark or the question marks. The baseline for sentence tagging would be a simple algorithm that reads down the string, and whenever a period or an exclamation or an question mark is found, assume that the previous sentence have ended and a new sentence begun. This is a good way until we run into spoken text and

casual writing such as email. There are two ways we have adopted in our code to overcome these problems. For the problem of speech endings, we have created a long list of possible punctuation strings. For the second problem, we listed a small group of terminal punctuations, and if only those appear in a string of punctuations, then call it the end of a sentence. We also run into the problem of headlines. Some text have headlines between in the `<TEXT></TEXT>` region, and those rarely have end-of-sentence punctuations. In order to minimize confusion to the segmenter, if sentence appearing on a line of its own and the `$sentence` token displays false, tag the whole line as a sentence. There may be exceptions to the rule, but this works very well for all newswire text. To see sample output from the sentence tagging module, please refer to table 3 and sample text in appendix.

4.3 Segmentation

4.3.2 MMS

The baseline for Chinese segmentation is the maximum matching string (MMS) model which is implemented in Erik Peterson's code downloaded from the web. Chinese read from left to right. So, all of the multi-character words can be matched by scanning the text from left to right, and applying a greedy algorithm. All possible word matches are compared with the words in a lexicon. Each loop of the algorithm a morpheme is added to the string to the right of existing string, and if the new string exists in the dictionary, the character is added to the string. If the newly formed string makes no sense, then the old string would be segmented out as one word, and the new character treated as the leftmost character of a potential new word.

Generic sentence:

Our test sentence: *Chinese agriculture is developing quickly.*

```
(zhong - central)
  (zhongguo - China)
    _ (zhongguonong - makes no sense, so nong is separated)
  _ (*_nongye - * agriculture)
    _ _ (*_nongyefa - make no sense, so fa is separated)
  _ _ (*_fazhan - *_development)
    _ _ _ (*_fazhanxun - make no sense, so xun is separated)
  _ _ _ (zhongguo_nongye_fazhan_xunsu complete sentence)
```

Figure 6 Basic application of Maximum Matching String algorithm

This method works fairly well for short strings of one or two bounded morphemes. But the most obvious problem is among words that does not progress in a one-morpheme-at-a-time fashion. Idioms are common in Chinese. Majority of the idioms are composed of four or five characters. Many times the first one or two characters might make sense as a word, but how do we take a leap from one or two character string to a four or five character string? The key is building differentiation into the wordlist. When a lexical dictionary is initialized for computation purposes, we will create two states for each string. One is “*this string exist as an independent word*”. The second state is “*this string may exist as an independent word, but only if you add more to it*”. We will also add a temporary variable that can store the characters which are uncertain as to whether they will be added to the previous segment. So when a string is looked up in the lexicon, we get one of three possible states: “don’t exist”, “add more”, or “this is a word”. The figure below illustrates the application of this algorithm.

Sentence with an idiom:		
(This is the plan to kill two birds with one stone)		
The idiom:		
(yijianshuangdiao - Killing two vultures with one arrow)		
Segment	State	Temp Variable
— — — (zhe_shi_yige_yi - this is a ...)	add more	(yi)
— — — (zhe_shi_yige_yijian - this is a ...)	add more	(yijian)
— — — (zhe_shi_yige_yijian - this is a ...)	add more	(yijianshuan)
— — — (zhe_shi_yige_yijianshuangdiao - this is a killing two birds with one arrow...)	word	(yijianshuangdiao) <i>*clear temp var</i>
— — — (zhe_shi_yige_yijianshuangdiaode)	don't exist	(de) <i>*clear temp var</i>
— — — — (zhe_shi_yige_yijianshuangdiao_deji)	don't exist	(ji) <i>*clear temp var</i>
***	***	***

Figure 7 Working with idioms in the MMS algorithm

This segmentation algorithm forms our baseline performance. But it will run into problems when the greedy algorithm cuts characters from other words, or bound free morphemes. An example of that can be found in the figure below.

Original sentence	(Thirty people attended the meeting)
Greedy MMS output	— — — — (Thirty ginseng... the sentence does not make sense)
Correct segmentation	— — — —

Figure 8 An example of a common mistake that greedy segmenter will always make. The region of ambiguity is displayed in bold.

Proposed solution to this common problem include incorporating a frequency model (in this case, “attended” will have a higher frequency of usage than “ginseng”), and utilizing part of speech information with a Hidden Markov Model, and applying heuristic linguist rules. We will examine some of those proposed solutions and their implementation in later sections.

Numerous MMS implementation for segmenting Chinese are freely available. The version of code that we downloaded and modified is from MandarinTools.com. The source code is written in PERL. The original wordlist consist of 23,000 words. We downloaded alternative wordlists that ranged from 50,000 to 130,000 words. Wordlists are a critical part of the lexical based segmentation process. The coverage of the wordlists, the definition of a word underlying the wordlist, and the character encoding all play a determining role on the performance and how detailed is the segmentation. But when we use a particular wordlist, we also have to keep in mind the part-of-speech tagging that we are doing in later module. So the wordlist used for segmentation has to agree with the part of speech wordlist. The ideal is to have a POS dictionary with very good coverage, so during our segmentation process, POS dictionary can be directly used.

MMS have built in numeral, time, and name identifier. The numeral and time segmentation is simply utilizes lists of time expressions, measure units, and Chinese numbers. In the greedy algorithm, if the string is completely formed from elements from one of the above list, then tag it as a time or numeral expression, respectively. During implementation and modification, we discovered there are numerous missing characters on these lists, and so we augmented list with our own. Personal name tagging is different. There is not easy solution to tagging personal names and the whole topic deserves a section of its own.

4.3.3 Personal Names

There are two categories for personal names: Chinese names and phonetic translations of foreign names. Techniques used in our segmentation tools to detect them are different, thus they will be presented in two separate paragraphs.

The identification of Chinese names comes after the recursive loop that does the initial segmentation. The wordlists does not contain any Chinese name, so presumably, all the Chinese names are presented as unbounded morphemes. The process that follows will attempt to identify which of the unbound morphemes look most like a name sequence, and then merge them. Chinese names have a finite list of surnames to choose from, and this list of surnames have changed little over recent history. Characters that can make up given names are not limited except for some particles that absolutely can't be part of a name. But the length of the given name is limited to two morphemes. The most frequent length of a whole name is three. Any lexical base segmentation program would first look for single surname character that is not bounded to larger strings. Then, if it is followed by a title (looked up on a list of possible titles), or followed by one or two unbounded morphemes, or preceded by a name predicate, the string is most likely a person. This simple algorithm is coded into Erik Peterson's MMS script. As an extra layer of precaution, the name identification subroutine also looks at a list of characters that frequently appear unbounded, and which rarely appear in names. This list includes a lot of particles, interjections, and prepositions.

The segment-then-merge approach of tagging names runs into problems when any part of the name characters can be bounded to boundary morphemes. An example of that is imbedded in following sentence:

(erzhong de wanglixue xiaozhang fayan - chancellor Wang Lixue of No.2 middle school speaks)

(wanglixue) is the name of the chancellor. But when the name is put together with the title (xiaozhang - chancellor), we can form the new word (xuexiao - school). A literature search revealed extensive study of this problem. One proposed technique

uses frequency models³⁸. When an unbounded surname morpheme is found (it did not mention the case where the surname is bounded), two morphemes to the left of the surname are looked at. A list of character frequencies built on a very large bank of Chinese names is used to determine if these two morphemes are “likely” to be part of the given name. The likelihood of this whole name string is balanced by the probability of the morphemes appearing alone (this include the surname), and the probability of the leftover morpheme appearing unbounded. The general conclusion of this study is that over 90% of the time, given name characters are unbounded. Of the ambiguous cases, the second morpheme is likely to be separated from the polysyllable word and put together with the surname, and the third morpheme is likely to remain in any polysyllable word. Another problem with name identification is abbreviation. In some literature, the appearance of the given name on its own is frequent. This is not the case for newswires, however, as most Chinese names are written out in full. In some editorials, the full name may be used once, and then only given name is used later in the article as a sign of familiarity. Segmentation tools have very hard time catching this. Post-segmenter-processing module for Alembic have a capability of remembering earlier mentions of the full name and make a decision on whether later strings contain characters that suspiciously match those of earlier mentions and tag them as personal names.

Handling foreign names is different from handling Chinese names. Foreign names do not have finite surname list or length restriction. Fortunately, foreign names are usually translations that derive their phonetics from a finite set of Chinese characters. This set of characters range from widely used to characters that specifically mimic a sound. The baseline approach to tagging foreign name is to find a string of these foreign character that can not be placed in polysyllable words with boundary characters and put them together to form a foreign name.³⁹ This is done in a way similar to the algorithm used to tag numerals and time. There is still the problem of telling a foreign person name from a foreign location name. This is currently an unresolved problem. However, majority of

³⁸ Large Corpus Based Methods for Chinese personal Name Recognition, *Journal of Chinese Information Processing*, 6(3):7-15

³⁹ Erik Peterson, 1996

the locations are followed with a location identifier, so the LISP post-processor is able capture false tags and convert them into the correct ones.

Surname	Foreign phonetics	Non-name	Numerals	Classifiers
...

Table 4 Sample list of the support wordlists used for segmentation. These are universal to MMS, Unigram, and Rule based system.

4.3.4 Heuristic Rule Based System

The previous sections on maximum matching algorithm described the simplest implementation of a Chinese segmentation code. But towards the end of 4.3.2, we gave an example of an ambiguous case where MMS will fail every time. Without a look ahead model, those ambiguity cannot be resolved. Prior research with look ahead models has ranged from a simple two word look ahead, to hidden Markov models over the whole sentence. In the end, the cost associated with building a very complex system, namely speed, becomes inhibitive with only a very small gain in performance. When the performance has reached a certain level, the solution provided by a complex system for one type of ambiguity will usually create another. So what is the balance point between performance and speed? A high number of research facilities have settled on three word look ahead models. A paper in published by Chen and Liu in 1992 detailed one such segmentation algorithm. It is simple enough that processing time is not dramatically more than the simple MMS. It incorporates 6 rules to determine what is the best segmentation. 5 of those rules were applied in our processing module. The rules are listed below. Their rank of importance go from A to E.

- A. Most plausible segmentation is the three word sequence with the maximal length. With a simple look ahead, find the three word segments that , together, achieve the longest length.
 - ABC_D_EFG (correct)
 - ABC_DE_F (incorrect)
- B. When there is more than one way to choose the three word segments of equal length, pick the word chunks with the smallest standard deviation for the word length.
 - AB_CD_EF (correct)
 - AB_C_DEF (incorrect)
- C. Normal words get higher priority than determinative-measure compounds. So if AB_CD_E and AB_C_DE are the choices, and C is a measure compound, then pick AB_CD_E.
- D. Pick the chunks with the high frequency monosyllabic words.
 - AB_CDE_F (F is low frequency)
 - A_BCD_EF (A is high frequency, then pick this one)
- E. Pick the chunks with the highest probability based on their part-of-speech information.⁴⁰

There is an implementation of this method available off of the web. The code is developed by Tsai for his Master's degree at University of Illinois. It was written in C and used BIG5 input and output. Our system used GB character encoding. So a PERL script was written to take GB input, convert it to BIG5 encoding, pass it to the segmenter, and convert the BIG5 output back into GB. Tsai's code was tested against the simple maximum string matching and, surprising, did not perform much better. While it resolved some ambiguities, it also created new ones. It was also lacking a good time and numeral and name algorithm. This problem is overcome by building the time and numeral and name subroutine similar to one found in MMS into the PERL shell script. This time, the performance was better than that of MMS, but still contained a fair amount of errors. From observing the two preliminary results, there is a proposal to add statistical model on top of rule A. We proposed that rule A combined with a simple statistical model might be sufficient overcome some of the problems and we only need to implement B and C to get significant performance improvement.

⁴⁰ Chen and Liu, 1992 COLING

4.3.5 Unigram Frequency Model

The third segmenter fields a statistical rule in place of the maximum length rule of the second segmenter. The rest of the rules are similar. This modification is aimed at resolving some errors made by the first rule. Specifically, to deal with uncommon words that might have a longer length than common words.

A. We will still find all the possible maximum string matching for three word segments. For example, for the string ABCDEFG, we have found the following ways to segment it into three word character sets.

1. ABC_DE_F
2. AB_CD_EF
3. ABC_D_EFG

In the original rule, third segmentation string would be picked because it is the longest. In the new method we will observe the number corresponding to frequency of occurrence for each word. Let's call that number α . We will also substitute a number η for the length of the word. η is a constant for each word length that increases non-linearly with the word length. For each Chinese word, we will introduce a weight $\alpha^\phi \eta$. ϕ is a set number depending on how much we wish to weigh the frequency factor. For each possible three word string, the weights of the words are added together and compared. If there are significant differences, either due to the frequency of occurrence or due to length of strings, we will pick the one with the highest weight. If difference is not significant, then we will resolve the conflict by picking the maximum string. If difference in weight is not significant and there are equal length strings, we will resort to lower level rules.

The frequency data for words is found in a list downloaded from the web⁴¹. There are huge flaws to this design. First flaw is the way the frequency data is obtained. The original segmenter with heuristic rules is used to scan through 17 million characters. The resulting segments are stored and counted to obtain the frequency. So the frequency count would be biased towards the mistakes that the rules-segmenter systematically makes, the same mistakes that we are trying to correct with segmenter that incorporates frequency data. Secondly, the wordlists used in the original rules segmenter does not agree with our segmentation guidelines. As a consequence, the wordlist with frequency data does not match the smaller wordlists that we would like to use to do our segmentation. We feel that the wordlist with frequency data does not follow “*smallest unit of meaning*”. Thirdly, the original rules segmenter does not have a personal name

⁴¹ <http://www.chinesecomputing.com/>

identification algorithm. So we suspect that the frequency data would contain frequency of characters from personal names as well, shifting the balance of segmentation towards to monosyllable “words”. In the end this trial at building a segmentation script is flawed, and will ultimately be discarded for other software with higher performance.

4.3.6 PKU OTS Software

Up till now we have explored different segmentation codes. We have also done modifications, and implemented our own with combined resources off of the web. But the efforts focus only on segmentation. In the end, we would have to do part of speech tagging as well. However, both of these capabilities are found in the OTS software from Beijing University. We do not have the source code to PKU segmenter. Thus we can only speculate on how their segment is implemented and give some examples of where their software performs well, and some examples of where their software fails.

PKU segmenter takes in an article or sentence without any extraneous tags and outputs segmented and part-of-speech tagged data. For some cases it performs rudimentary phrase tagging. We suspect that their segmentation and part of speech are implemented on two different levels, just like our system: segmentation on the first level, and part-of-speech on the second level. With the most basic testing and deduction, we know their segmentation algorithm incorporates a look ahead model. In cases where we fed sentence with ambiguities, it almost always came back with the more common word as the segmentation result. So we suspect that they use a frequency model to decide between conflicting words. They also have a fairly powerful personal name tagger. Behind this name tagger is the same algorithm surname-given name algorithm we have discussed in section 4.3.4. But it is more complex because there may also be a frequency model build into it. The following table illustrates some of the sample data that caused us to come to these conclusions.

Input sentence	Output sentence	Deduction
	/m /n /v /u /n /w	There is look ahead capability.
	/n /v /n /d /a /w	Frequency of occurrence is used when disambiguating conflicts.
	/nr /p /r /v /u /m /q /w	Extensive name tagging capabilities.
	/nr /v /n /v /w	Frequency model built into the name tagging system.

Figure 9 Input and output of the PKU segmentation &POS processor.

The inner workings of the part of speech tagging of the PKU software are harder to deduce from the output. However, we believe that it is also lexicon based, and have a frequency model. The part of speech assignments make systematic mistakes that we plan to correct using LISP rule based module, such as making nouns into verbs, or verbs into nouns.

The PKU software also attempts to do basic level named entity tagging, although the named entity tags in the output is more of a distraction than anything we can use to satisfy the named entity task. The way they do is very rudimentary: taking a large list of organization and locations from the manually tagged People's Daily data and checks it against all the input. For example, the software would be able to catch (meiguoguoofangbu - US Department of Defense) and tag it as one organization, but would tag (Australian Department of Defense) separately as a location and a noun. We build a LISP level NE task system based only on part-of-speech information and segmentation, and will not rely on their phraser.

As a final note in describing the PKU software, it has different input/output requirements than Alembic system. In order to integrate this software to our system, a shell program written in PERL is built. This script takes sentence tagged articles and pass each individual sentences to the PKU segmenter, then take the output from PKU segmenter and format it into the LEX form that Alembic can read and use. Currently

such a system is implemented only for their web-based demo, where we pass individual sentences to their demo site and retrieve the information we want from the resulting html page. We have not yet obtained their software license, so we don't have their source code or the actual program running on our computer, thus our testing has flaws. One of the biggest problems is word limitation built into their web demo. We cannot input more than 50 characters at a time. This cause the break up of some longer sentences along the 50 character mark, which potentially can throw off their segmentation process. The main reason we have tested their software extensively is that their software uses the same part of speech tags as their freely available tagged and segmented People's Daily corpus. This large corpus provides valuable training for the phrasing task, the LISP segmentation module, and POS module. Since the NE phrasing task is done with the assumption that the part of speech information will follow the same format as the output of the PKU segmenter, we are very keen on obtaining their software and making it a critical or optional part of our system.

4.3.7 Rule Based Alembic Segmentation Software

One more avenue that we have explored in brief is a rule based segmentation module based on LISP. It is peculiar because in this system we are trying to merge characters rather than separating them. All of the sentences get parsed into character sequences where each morpheme is treated like a word before being passed onto the segmentation module. The segmentation module then uses a large word list to merge characters together to form words. This process is seriously explored only as a machine trained rule sequence. Training took place with 200,000 words from People's Daily, and after learning a sequence of 160 rules, the performance level was 83% for training corpus and 80% for a smaller testing corpus. Because the performance level was so low, we did not explore it further. But with the licensing of a larger and more comprehensive wordlist with POS information, we maybe able to go back to this original concept and attempt it again. More will be presented in the discussion section.

4.3.8 LISP Part of Speech Tagging

We have presented in sections 4.3.2, 4.3.3, 4.3.4, and 4.3.5 how we have approach the segmentation problem independent of the part of speech problem. In this section, the LISP based POS tagging solution is presented in finer detail. We have decided to use Alembic rules environment to complete the POS tagging because: 1. it is flexible environment, we can easily change the wordlists, add wordlists, and delete them without changing much of the hard code, 2. it is trainable, and thus we can automatically generate rules by giving the machine all the resources that a human rule-writer would have and let it run over the large training corpus.

For the benefit of training, we have adopted the same set of tags as those provided by Beijing University. The first step is to develop a POS wordlist to aid training and tagging. This is done by collecting all the words from the People's Daily data, and separating the words into files categorized by their part of speech information. One word may appear in several files since they may be used in several POS positions. The part of speech information derived from the People's Daily data is presented in the following figure:

POS	Total	1 Char	2 Char	3 Char	4 Char	Other
Ag	321	321	0	0	0	0
Mg	7	7	0	0	0	0
vd	528	17	501	10	0	0
nr	35262	18537	13927	1415	623	760
a	34473	12834	21577	54	8	0
ns	27890	0	18584	7297	1354	655
nt	3573	0	7	3118	416	32
b	8715	1913	6120	646	26	10
c	25479	18923	6228	176	152	0
d	47811	29845	16846	1117	3	0
vn	42734	127	41747	857	3	0
e	27	23	4	0	0	0
ad	5933	496	5435	0	2	0
nx	452	112	49	133	50	108
f	17201	12627	4499	75	0	0
h	47	47	0	0	0	0
nz	3700	0	2090	1034	437	139
i	5033	0	0	8	4966	59
Bg	8	8	0	0	0	0
j	10323	3723	3896	2169	518	17
k	928	928	0	0	0	0
Dg	135	135	0	0	0	0
l	6015	0	26	686	5104	199
n	236810	16846	186882	26800	5258	1024
an	2837	68	2769	0	0	0
o	70	9	51	1	9	0
p	39925	36339	3585	1	0	0
q	24244	22048	1909	235	40	12
r	32327	13800	18395	132	0	0
s	3836	0	3655	179	2	0
t	20680	0	14019	3579	177	2905
Ng	4562	4562	0	0	0	0
u	74829	74363	466	0	0	0
v	184764	64953	118616	1176	17	2
y	1889	1857	32	0	0	0
Rg	10	10	0	0	0	0
z	1391	0	963	217	210	1
Tg	496	496	0	0	0	0
Vg	1749	1749	0	0	0	0

Figure 10 PKU POS of speech and the number of words associated with the POS. The character columns represent the number of characters in each word.

There are a total of 50,000 words. This covers most of the common words use in Chinese. However, this is not the complete list. The complete list would have close to double the amount of word coverage. Licensing for the complete list is still in the process of negotiation. In the meantime, the derived partial list is used to train the POS tagger.

In training, we used the People's Daily's data. Purist would argue that we should not be using the same data as the data from which the wordlists are derived. From another point of view, we are trying to tailor the rules for the situation where we have a complete list instead of just a partial list. Theoretically the complete lists should have a very good coverage. Thus, using the derived list to aid in tagging the corpus from which it is derived from is a time saving measure. When we get the licensing for a good part of speech dictionary for Chinese, we will not have to learn the rules over again because we have much better coverage for a broad corpus selection. More about the part of speech results will be presented in the results section.

4.4 Phrasing system in Alembic and OTS software

Our efforts for segmentation have been comprehensive. We have researched into and tested several systems that we might consider as part of the final product delivery for ACE Chinese. However, we are still in the process of evaluating our options, and have built a phrasing system fairly independent from the segmentation process. Our segmentation effort and part of speech tagging largely utilized the Beijing University's hand annotated People's Daily data⁴². Thus, the output from the segmentation and part-of-speech effort uses the same set of part of speech tags as Beijing University⁴³, and has the same segmentation guidelines. Our phrase training will be done with the same hand annotated data, and be based on the PKU POS tag set.

Ideally, we would have segmentation and part-of-speech exploration completed first before going on the phrasing. This way, the results from the phrasing module would present the complete system results up to that step of the process. But time and resource restraint meant that we had to pursue phrasing and segmentation/POS tagging in parallel. So the training and results are all based on perfectly segmented and tagged corpus. To get a sense of the final results up to NE task, we would have to multiply the final score of

⁴² See Appendix E1

⁴³ See Figure 5

the segmenter and POS tagger with the final score from the phrase task. There are also benefits from building a system like this. As we explored NE phrasing, we develop a clearer picture of the patterns underlying name, organization, and location phrases, unclouded by systematic mistakes generated from the segmenter and POS tagger.

4.4.2 Training

We took People's Daily data (one large file) and converted it into a format the Alembic phrase rule learner can read and saved into a database. The total database is around 3,000 files, or 3,000 articles. The file was named with by the date of the article, the section number to which it belonged, and the article number within the section. Then we cat'ed the 3,000 files into 10 large files and 10 small files. Those 10 large and small files are used for training and testing, respectively. To run the trainer, Alembic system is first loaded into a LISP interface. Then we load all the resources that the Alembic system is asked to use when learning, such as wordlists and list of tags. Finally we invoke the learning process at the LISP interface.

The first training pass we did was cold training. All that was given to the system was part-of-speech tagged corpus. The machine had nothing else to work with. In general, the rules that it developed looked at the part-of-speech tags, and suffixes of Chinese words for indicators of name, organization, and location. The second pass of training, we gave the machine wordlists to work with. The wordlists included the surnames, POS wordlists, foreign characters, location names, and other lists. This time the system performed better. However, it was clear that it did not know how to use the full potential of the wordlist that was provided to it. In the third pass, we separated the personal name, organization, and location, and trained the machine on each one of them individually. This is to see which area needs to be working on the most to achieve better over all scores. Results from each of these pass and the section conclusion could be found in the results section.

Supplementary wordlists	Description	Origin
Capitals.txt	List of national capitals	Translation of English wordlist
Classifiers.txt	List of Chinese classifiers	Manually compiled by linguist
foreign1.txt & foreign2.txt	List of Chinese characters used for phonetic translation	Compiled from MMS, MET2 data, BBN data
Locations.txt	List of Chinese and foreign locations	Compiled from MET2 and BBN data
Nations.txt	List of nations	Translations of English wordlist
Organizations.txt	List of organizations	Compiled from MET2 and BBN data
Surname.txt	List of surnames	Compiled from MMS, and People's Daily
Title.txt and Title2.txt	List of titles in Chinese	Manually compiled by linguist
us_states.txt	List of US States	Translations of English wordlist
zh_provinces.txt	List of Chinese provinces	Manually compiled by linguist

Figure 11 List of the supplementary wordlist used for machine training. See Appendix B5 for translation code.

4.4.3 Testing and scoring and error tracking

The Alembic system works with rules. Each rule is applied over the whole corpus, and no rule is ever repeated. The general trend is to go from broad rules to smaller rules. The rule learner functions much the same way. It has a list of patterns that it will look for. For each rule that it generates, it will attempt to achieve the maximum gain in performance out of the list rules that it knows how to generate. It doesn't have any ability to anticipate future rules. For example, sometimes it is more beneficial to tag the specific things first, so that less general rules can be written in the future to avoid over

tagging. The training system would almost always produce a smooth curve going from the broad rules to specific rules. Eventually it will reach a quasi plateau, when rules that it generates are so specific to the training corpus that it can't advance the performance on the testing corpus any more. We have graphed one training process below to show this curve. An example of the learned rules can be found in Appendix B2.

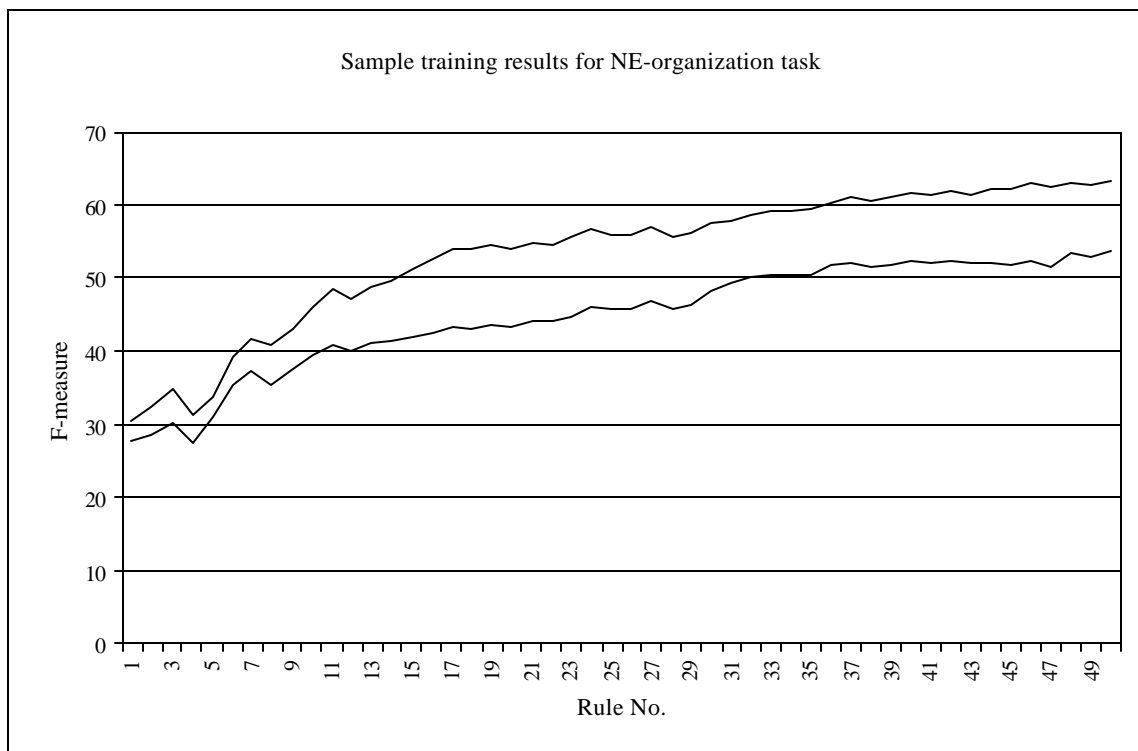


Figure 12 This shows an example of the training for organization tagging. Upper line represents performance on training data. Lower line represents performance on testing data.

The measurement we used for scoring is a unit called the f-measure. F-measure is the average between precision and recall. Recall is the percentage of phrases that we have correctly tagged out of all the correct phrases. Precision is the percentage of correctly tagged phrase out of all the tags we have used. We can have a very high recall because we have tagged almost all the phrases, but still get a very low f-measure because phraser did a lot of extra flagging and the precision level is down. The goal is to achieve both high precision and high recall: produce a lot of tags and make every tag count. The f-measure is what the machine looks at when learning.


```

* * * SUMMARY SCORES * * *
-----
          POS  ACT | COR PAR INC | MIS  SPU  NON | REC PRE UND OVG SUB ERR
-----
SUBTASK SCORES
enamelx
  organizatio  801  934 | 684  0  0 | 117  250  0 | 85  73  15  27  0  35
  person       0   0 |  0  0  0 |  0   0  0 |  0  0  0  0  0  0
  location     0   0 |  0  0  0 |  0   0  0 |  0  0  0  0  0  0
  other        0   0 |  0  0  0 |  0   0  0 |  0  0  0  0  0  0
-----
ALL SLOTS    1602 1868 |1292  0  76 | 234  500  0 | 81  69  15  27  6  39
              P&R      2P&R      P&2R
F-MEASURES          74.47      71.19      78.06
Under Subtask scores, Enamelx, Organization
POS: Number of real phrases in the corpus
ACT: Number of tags that the rules generated
COR: Number of correctly tagged
MIS: Number of missed phrases
SPU: Number of spurious phrases

```

Figure 14 Sample output from scorify program: *.scores file output.

```

Document 1998010103009
TAG      TYPE TEXT KEY_TYPE      RSP_TYPE      KEY_TEXT      RSP_TEXT
-----
ENAMEX   cor  cor  ORGANIZATION ORGANIZATION "      "      "      "
ENAMEX   cor  cor  ORGANIZATION ORGANIZATION "      "      "      "
ENAMEX   cor  cor  ORGANIZATION ORGANIZATION "      "      "      "
ENAMEX   mis  mis  ORGANIZATION "      "      ""
ENAMEX   mis  mis  ORGANIZATION "      ""
ENAMEX   spu  spu  ORGANIZATION ""      "      ""
ENAMEX   spu  spu  ORGANIZATION ""      "      ""

```

Figure 15 Sample output from scorify program: *.report-summary file output.

The information in figure 15 is very valuable in helping us generate manual rules. More about the process of manually generating rules and testing them is presented below.

4.4.4 Compiling the frequency data

A detailed list of the results for the machine learned phrase tagging system will be presented in a later chapter. To summarize, the machine was unable to achieve very high performance for tagging organizations. So we had to find experiment with manually

writing rules to tag organizations. A sample rule sequence is presented in the appendix section B3. The systematic approach to writing good rules for tagging organization is to find the most frequent patterns and write rules based on those frequently occurring patterns. A script was written to collect all organizations and their part-of-speech information. Then the part-of-speech patterns that had more than 50 hits were recorded. The sample results are presented below.

```

<LEX POS="n"> </LEX><LEX POS="n"> </LEX><LEX POS="vn"> </LEX><LEX POS="n"> </LEX>
<LEX POS="nt"> </LEX><LEX POS="n"> </LEX>
<LEX POS="n"> </LEX><LEX POS="j"> </LEX><LEX POS="j"> </LEX>
<LEX POS="n"> </LEX><LEX POS="spos"> </LEX><LEX POS="n"> </LEX>
<LEX POS="ns"> </LEX><LEX POS="j"> </LEX>
<LEX POS="ns"> </LEX><LEX POS="j"> </LEX>
<LEX POS="n"> </LEX><LEX POS="n"> </LEX><LEX POS="n"> </LEX>
<LEX POS="n"> </LEX><LEX POS="j"> </LEX>
<LEX POS="nt"> </LEX>

```

Figure 16 Sample of all the organization collected by scanning through the People's Daily data.

337	nz_n_
423	n_n_
158	ns_n_n_n_
130	ns_ns_n_
216	ns_j_
117	n_n_n_
81	nt_j_
92	ns_n_vn_n_
1364	ns_n_
57	n_n_n_n_
160	nt_n_
89	ns_nt_
60	ns_ns_n_n_
130	j_n_
64	ns_vn_n_
70	ns_nz_n_n_
2487	nt_
135	ns_nz_n_
576	ns_n_n_
342	n_j_
69	nz_n_n_

Figure 17 The most frequently occurring patterns to organizations.

4.4.5 Generating wordlists to aid tagging sequences

In the first and second pass of the machine training, we have not achieved the desired results for NE task. The machine learning sequence was still geared towards English. However, even as the machine can not achieve the results we are aiming for, it is already noticing some particular traits of Chinese such as the suffixes that denote organization. This “knowledge” is not applied in a generalized way, thus it is unable go above a certain level of performance. We seek to take advantage of these “islands” and expand upon them tag organization, and achieve higher level results with locations. This effort resulted in a couple of key lists that is used for manually generated rules. There are the base suffix list for organizations and locations which is collected from numerous manually tagged corpus such as MET2⁴⁴, and PKU’s People’s Daily. Then we have a list of frequent words that contain the suffix for organizations but are actually not organizations. The addition of these wordlist resulted in marked improvements for the manual rules. In fact, each additional entry into the wordlist results in a slight but noticeable improvement at this early stage. Below are sample entries from the lists.

Locations	Translation (example)	Organization	Translation
	Harbor (Hong Kong)		Company
	Island (Hainan Island)		Department
	River (Yangtze River)		Central
	Lake (Kuiming Lake)		Agency
	River (Panama Canal)		Group
	Stream		Organization
	Village		Department
	Gorge (Three Gorges)		Government

Figure 18 Sample list of Organization and Location names.

⁴⁴ See Appendix E3

Non-location	Translation
	Society
	Officer
	Meeting
	Q&A Meeting

Figure 19 A list of frequently occurring words that include the organization suffix but are actually not organizations.

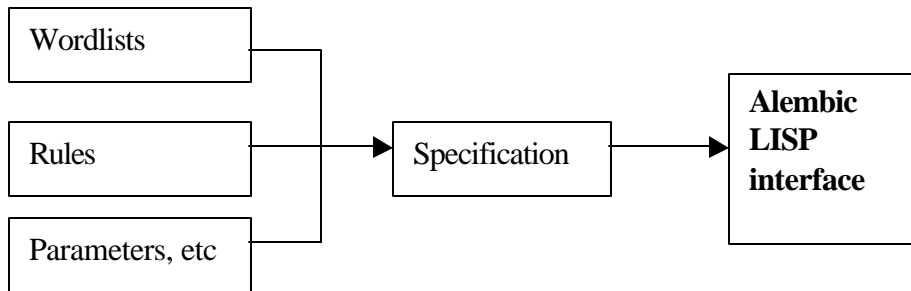
4.4.6 Manually Generated rules

Our manual rule generation focused on the poorest performing part of the NE task. It seeks to utilize all the information and resources that we have collected in the most effective way. The process of writing rules is a recursive process. First we write the rules. Then the rules are implemented over our test corpus. Next, we score the test corpus, and check both our results and the report-summary, with the missing and miss-aligned tag report. The report is carefully read through, and new items are added to the wordlists and new rules are written.

There are two ways to approach tagging. One way is to identify islands within the phrases, and then expand left and right accordingly. The other way is to find the POS tags or words around the organization or location phrase that provide good hints. We achieved our current results by mainly operating under the island principle. Future work would probably require using both approaches.

The first rule for tagging organization is a rule to capture all the words that end with the organization suffix. This action provides the “islands” off of which phrases can be expanded. Then we write a rule to exclude those words that definitely cannot be an organization but was tagged because it had the organization morpheme. Next we use the frequency data with part of speech information, going from the most specific order of POS tags to the most general. Finally we get rid of all the “islands” that did not fit under any of the organization POS format. The written rules for tagging organization are attached in Appendix B3.

Running the rules required developing a specification file to call on the rules and to load the necessary wordlists. A sample spec file is attached in the Appendix B1.



5. Results

There are two sections of results. The first section is a qualitative analysis of the segmentation and POS tagging phase. The second section is a more quantitative analysis of the NE task. The results from phase 1 and 2 are separate from each other.

5.2 Results from phase 1

5.2.1 Sentence tagging

We achieved close to 100% tagging capabilities for sentences from newswire. A sample of this output can be found in Appendix A1. One of the reason for the high performance is because the People's Daily and Xinhua New Agency data are highly predictable and structured. The sentence tagging program had not been tested against data which had false return carriages or bad punctuation. Another reason why we had such good results is because we had not made the distinction between headlines and regular sentences. In a more formal evaluation, headlines have to be picked out from all the sentences. This task presents less of a challenge in English because headlines are simple capitalized sentences without a termination punctuation mark. For Chinese, which have no capitalization, this task is a project by itself. Since we are seeking to build an end-to-end NE system, not that much time could be spend the working on distinguishing headlines from regular sentences. Thus they are all treated like regular sentences.

5.2.2 Segmentation

The output from MMS, heuristic rules, and unigram segmenters can be found in Appendix A2, A3, A4, respectively. The simple MMS segmenter downloaded from MaradarinTools.com is converted to output text forms that Alembic system can read. It is also outfitted with better numerals, surnames, foreign name characters, and measure words. The adaptation of Peterson's MMS segmenter performed surprisingly well,

especially when compared with the preliminary results from the heuristic rules segmenter with the look ahead model from Tsai. The biggest draw back to the rules segmenter is that it did not have a good numeral or personal name identification algorithm. Many of mistakes assigned to the heuristic rules segmenter are solved by implementing a script on the output of the segmenter to merge names and numerals with an algorithm similar to the one found in the modified MMS segmenter. However, even with the script, there are surprising mistakes that the rules segmenter makes. Overall, aside from personal names and numbers, it was able to solve more problems than it created.

Sample segmentation result for:	(loose cargo container ships)
Correct segmentation output	
<LEX> </LEX><LEX> </LEX><LEX>	
MMS segmentation output (incorrect because it was greedy)	
<LEX> </LEX><LEX> </LEX><LEX>	
Heuristic rules segmentation output (correct because of minimum variance rule)	
<LEX> </LEX><LEX> </LEX><LEX>	

Figure 20 Comparison result for MMS and Heuristic rule segmenter. Heuristic rule made the correct segmentation judgment.

The implementation of the unigram model proved to be very difficult. The reason is the frequency list that we are using. Some of the potential problems have already being discussed in the methods session, mainly the ill match between the wordlist that contains the frequency data and the wordlist that we are using to build the segmenter. The result is that unigram model for segmentation functions just like the heuristic rules segmenter. The frequency for the word segments did not vary enough to make a real difference in most of the segmentation ambiguities. However, we will soon acquire a much better frequency wordlist from University of Pennsylvania. This wordlist is machine compiled and manually edited by linguistic researchers, and mostly agree with our segmentation guidelines. Integrating this wordlist into the unigram frequency model segmenter should produce results more in line with the expected improvement.

Even though the theories behind the segmenters we built or modified represent the best ideas of the field right now, our segmentation results still lack behind that of Beijing

University's segmentation software's⁴⁵. The OTS software seems to have a better lexicon and a more sophisticated way to disambiguate conflicts. It also seems to have a better frequency wordlists, which helps at most times, and throws the segmenter off at some rare instances. See figure 9 for specific example. But unless we make significant improvements to our lexicon, and add a good LISP level correction module (see discussion), the software of choice is still PKU's segmentation/POS program. This view is reinforced when we take a closer look at results from training in part of speech.

5.2.3 Part of speech tagging

Segmentation cannot be viewed independently from part of speech. Without a solid part of speech tagging module our segmentation effort is fruitless when we reach the phrasing level. We trained a LISP part of speech tagging module over a segmented corpus of the People's Daily data. The training took around 3 days. The learner generated close to 100 rules. The final result on the training data was 83.7% f-measure, and 79.2% f-measure on the testing data. This is significant because with this level of performance on machine generated rules, we believe manual correction or manually written rules would significantly boost performance.

We do not have a quantitative measurement for the performance of the OTS software. From observation, it made some systematic mistakes. Most of the mistakes are not the result of miss-segmentation; they result from conflict between part of speech assignments. An example of that mistake is present in the figure below.

POS Tagging Result for OTS software:	(Your criticism is correct)
Correct POS tags:	
/r /u /vn /d /a /w	
Miss tagged as:	
/r /u /v /d /a /w	

Figure 21 POS mistakes made by OTS software.

⁴⁵ See Appendix A5

Such examples of mistakes are few though. They are usually due to a missing word in the part of speech lexicon. The general performance is higher than the machine learned rules. This will continue to be the case until we have better resources on which built better rules.

In conclusion, the result showed us that we can build a rudimentary segmentation and part of speech tagging system based on the resources that we have derived or obtained off of free web sites. The performance of this system is on par with capable software in the industry right now. Still using the same framework, we can boost this performance by a huge amount if we can get better wordlists, and manually generate rules for the part-of-speech tagging module. Despite this confidence, we also acknowledge that the current system is under performing compared the Beijing University's OTS segmentation and POS software. Thus, the Beijing University software should be licensed. There will be two main benefits. One is we already will have a state-of-the-art software to complete phase 1 of this project. Two is we will have a very complete part of speech dictionary and a finely segmented wordlist, and this will allow to a more complete exploration of our own approaches to phase 1.

5.3 Results from phase 2

5.3.1 Machine Learned Rules for NE Task

No. Rules	Tags	F-measure	Resources
50	NT NS NR	65.27%	None, no wordlists were provided. Cold train session.
80	NT NS NR	84.91%	Part of speech wordlists derived from People's Daily
150	NT NS NR	87.58%	POS wordlists
70	NT NS NR	86.98%	POS wordlists Supplement lists Suffix lists
10	NR	99.91%	POS wordlists Surname lists Foreign phonetic characters
10	NT	41.50%	POS wordlists
50	NT	53.80%	POS wordlists
50	NT	57.24%	POS wordlists Org morpheme list
40	NS	88.90%	POS wordlists
50	NS	90.50%	POS wordlists Loc morpheme list

Table 5 The training results for NE task.

The above table shows the main results from training log for NE task. In the first row we showed an f-measure of 65.27%. This is training without any resources. All of the later training would take advantage of the wordlists derived from the PKU data. The f-measure shown in all cases are results for the testing data, not the training data. Except for the first learning session, the training data always has a 3 to 4 percent advantage over the test data. This is attribute to the fact that the wordlists are derived from the training data, plus the rules are tailored specifically to boost performance in the training data.

There are some clear trends in the numbers on the table. We will first focus on the processes that tried to learn rules for tagging all the named entities. In general, the performance goes up as the number of rules is increased, unless additional resources are added. We can see this in the improvement of f-measure from row 2 to row 3. There is a 2.67% increase in performance as the rules are almost doubled from 80 to 150. As the machine reach the limits of its intelligence, each additional rule only result in a very small increase in performance, sometimes not at all in the testing data. Thus we see the clear plateau.

With the addition of new resources such as wordlists, the performance goes up even if the number of rules remains the same. This is illustrated in the 2.07% increase in performance from row 2 to row 4. The number of rules decreases from 80 to 70, but this is more than offset by the addition of useful new resources.

After reviewing the tagged results from the first sequences of learning, we start training the machine with one tag at the time. Immediately, it is clear that the encouraging results we have are mostly due to the correct part of speech information from segmentation. Personal names accounts for majority of the NE tags (70%). An independent training on personal name achieved 99.91% f-measure within 10 rules. But we know personal names are already pre-tagged as NR. The surname and the given names are separate. But a simple rule that put these together would have resulted in a very high score. The second most prevalent named entity are locations. They are also pre-tagged in the segmentation process. Majorities of the locations are single words with NS as their part of speech. When the trainer learns how to independently tag NS, the first rule it generates is to convert the NS part of speech tag to NS phrase tag. This immediately takes its f-measure to high 70s. The recall is very high at that point. But precision is fairly low because it over flags all the locations that are used inside organization names. The rules following would mainly try to correct the over flagging. We can see that the location tagging learner achieved 90.50% f-measure in 50 rules.

The only named entity task that the machine cannot “cheat” on is organizations. There are organizations that are only one word, but those are rare. Thus the learner cannot use the part of speech information directly. The result is a punishing score on the organization tagging task (57.24% with 50 rules). List of morphemes that denotes some kind of organizational structure was generated and given to the learner as part of its resource. This only resulted in a slight increase in performance. The learner does not really understand how to use this list to the maximum advantage. Manually generate rules that really take advantage of the wordlists and patterns can achieve high results with minimal rule sequences.

5.3.2 Improvements with Addition of Custom Wordlists and Manually Generated Rules for NE Task

18 rules were written for the NE task of organization tagging. Please see Appendix B3 for the complete list. The way these rules are written is discussed in the last section of the methods section. They achieved significantly higher results. By the Alembic scorer, the f-measure for organization was 73.97%. Scoring using the program *scorify* produce an f-measure of 74.47%. The report-summary showed all the correct, missed, and spurious tags. There were a lot of spurious tags, which keeps the f-measure low. However, linguists that looked at the spurious tags determined that at least 30% of them are actually legitimate. It is a case of disagreement over NE guideline with the People’s Daily annotated data, which seemed to be inconsistent in many respect. The following table shows the kind of organization that they would tag in one article but would ignore in another.

Organizations	Translation
	Waste treatment plant
	<i>Banyanpu</i> College
	Representative Body for Committee on Peace
	Beijing Military Unit
	Literature Studies Society
	Central Military Task Force

Figure 22 A partial list of tags marked as errors by the scorer because they are not tagged as organizations in the PKU's People's Daily data.

The result shows clearly that we can achieve high performance for NE tasks in Chinese using the Alembic rules environment with the right wordlists and rule sequences. Noticeable improvements come with every extra addition to the still-limited wordlist. To avoid spurious tagging, a list of non-organizations with organization morphemes is generated to aid the rule writing process.

6. Discussion

6.2 Separation of phase 1 and phase 2

When name entity detection is attempted in the Alembic LISP based system, correctly tagged corpus from People Daily was fed into it. It is point of controversy because this corpus is perfectly tagged. How will we know the real performance of the NE task? The reason why perfect data is passed onto Alembic system for training is a combination of time constraint and the effort to train a pure named entity tagger without error correction for the segmentation and part-of-speech.

We wanted a modularize system. A post process module is envisioned to make corrections to the segmentation and part of speech tagging process. Our NE rules will be after the correction module, thus it shall be pure to the NE task. Whatever error correction could be done in the LISP rule based environment would have been done already. To build a modular NE phraser, the best data to pass it is the correctly tagged data.

So what is the real performance of the system up to this point? There is no quantitative answer to that question. It will be answered as part of our future work. But it does not belong in this project report.

6.3 Segmentation

Can segmentation be done using a rule based system? The initial results are not good. The instinctive answer would be no, because rules are applied over the whole corpus, and they do not facilitate scanning strings in a localized area. However, there maybe a curious to way to implement a rule based segmenter, and it would actually use frequency information and have a strange form of “look ahead”.

The corpus to be segmented will be passed onto the segmenter completely chunked into individual morpheme (the same format we are using to pass to Alembic segmenter right now). We can arrange the frequency wordlist according the frequency of words,

starting from the most frequent to the least frequent. Then we will have one rule that will go down the wordlist and scan for each word in the corpus. If it finds words or morphemes that can be merged to form the word from the wordlist, it will merge them. It will run through each word in list and attempt to find a match. But it will only call it a match if the new word can be formed without tearing old bond apart. This way we will always be tagging the most frequently occurring words whenever there are conflicts in segmentation. We can also try the same segmentation algorithm, but arrange the wordlists according to length. This will be explored in future work.

6.4 Post-process for OTS module

In testing the Beijing University software we noticed some systematic errors that it makes. These errors can be corrected using module in Alembic. One of the common errors is miss-tagging. They result from oversight when building the lexicon. One example of this is presented in figure 21. An rule that can correct this particular mistake is:

```
(def-phraser-rule
  :anchor      :lexeme
  :condition   (:wd :pos :|v|)
              (:lwd-1 :pos :|u|)
  :action      (:set :wd :pos :|vn|)
)
```

This is just one of example of many errors that we can attempt to correct in Alembic. This should boost the performance of our segmentation and POS tagging.

7. Conclusion

The mass of electronic information grows exponentially as more of the world becomes wired. Intelligent content extraction software that can help individuals and organizations make mission critical decisions based on this growing maze of information is becoming a necessity. A part of this effort focuses on building a multilingual information extraction system. This Master's project is a comprehensive study of the preliminary requirements to outfitting MITRE's Alembic Natural Language Processing tool with Chinese Automatic Content Extraction capabilities. It included the study of segmentation, part-of-speech tagging, and named entity phrase tasks.

This project has successfully explored the essential task of segmentation and part of speech tagging for Chinese. We can draw the following conclusion from our results:

1. Existing segmentation algorithms have been implemented for the pre-Alembic process based on code written and modified through this project. Results are on par with industry standards.
2. Replacement of the wordlists and resources within the existing segmentation/POS system can result in significant improvements.
3. State-of-the-art software from external sources is shown to work well with the Alembic environment. Should we choose not to develop our own segmenter, license for OTS software can be readily obtained.

This project has successfully explored the Name Entity task. We can draw the following conclusion from our efforts:

1. High performance can be achieved through a combined effort between segmentation, POS tagging, and rule based phrasing despite the unique problems facing Chinese text processing.
2. Manually generated rules and wordlists based on scientific analysis of named entities in Chinese can perform significantly better than simple machine learned rules.

8. Bibliography, Reference, and Resources

8.2 Web based resources

<http://www.mitre.org/resources/centers/it/g063/alembic.html>

<http://www.mitre.org/resources/centers/it/g063/workbench.html>

<http://www.icl.pku.edu.cn/nlp-tools/segtagtest.htm>

<http://www.icl.pku.edu.cn/Introduction/corpus tagging.htm>

<http://www.MandarinTools.com>

<http://www.geocities.com/hao510/mmseg/>

<http://www ldc.upenn.edu/ctb/>

<http://www.chinesecomputing.com/>

8.3 Academic papers and publications

Benjamin K. Tsou, Hing-cheung Ho ... , A Knowledge-based Machine-aided System for Chinese Text Abstraction, In *Proceedings of the COLING 1992*, Page 1039-1042

Benjamin L. Chen and Von-Wun Soo, An Acquisition Model for both Choosing and Resolving Anaphora in Conjoined Mandarin Chinese Sentences, In *Proceedings of the COLING 1992*, Page 274-357

Chang, Chao-Huang, and Cheng-Der Chen, A study on integrating Chinese word segmentation and part-of-speech tagging. *Communications of COLIP*, 3(2), December 1993

Chao, Yuen Ren. 1968. *A Grammar of Spoken Chinese*. University of California Press, 1968

Chen. Aitao, Jianzhang Hu, and Liangjie Xu. 1997. Chinese text retrieval without using a dictionary. *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Philadelphia, Philadelphia, July 1997

Chu-Ren Huang, Keh-Jiann Chen, A Chinese Corpus for Linguistic Research, In *Proceedings of the COLING 1992*, Page 1214-1217

Huang, Chu-Ren; Chen, Keh-Jiann; Chang, Lili; and Chen, Feng-yi. 1997. Segmentation standard for Chinese natural language processing. *International Journal of Computational Linguistics and Chinese Language Processing*. 2(2) , 47-62

Jerry Norman, *Chinese*, Cambridge University Press, 1998, Page 152-180

Jerome L. Packard. *The Morphology of Chinese: A Linguistic and Cognitive Approach*. Cambridge University Press, 2000

Jian-Yun Nie, Marie-Louise Hannan, Wanying Jin, Unknown Word Detection and Segmentation of Chinese using Statistical and Heuristic Knowledge, October 1995

Jones, Russell. 1997. *Chinese Names: The Traditions Surrounding the Use of Chinese Surnames and Personal Names*. Pelanduk Publications, Selangor Darul Ehsan, Malaysia, 1997

Jyun-Sheng Chang and Andrew Chang, A Statistical Approach to Machine Aided Translation of Terminology Banks, In *Proceedings of the COLING 1992*, Page 921-925

Jyun-Shen Chang, Shun-De chen, Ying Zheng, Xian Zhong Liu, and Shu-Jin Ke, Large Corpus Based Methods for Chinese personal Name Recognition, *Journal of Chinese Information Processing*, 6(3):7-15

Karen Spark Jones, Automatic Summarization: Factors and Directions, In *Advances in Automatic Text Summarization*, The MIT Press 1999: Page 1 –14

Keh-Jiann Chen, Chinese Sentence Parsing, *Presentation Slides for COLING 92*.

Keh-Jiann Chen and Shing-Huan Liu, Word Identification for Mandarin Chinese sentences, In *Proceedings of the COLING 1992*, Page 101-107

Kwok, K.L. 1997. Comparing representation of Chinese information retrieval. *Proceedings of the 20th Annual International ACM SIGIRE conference on Research and Development in Information Retrieval*, Philadelphia, Philadelphia, July 1997

Liang-Jyh Wang, Wei-Chuan Li, And Chao-Huang Chang, Recognizing Unregistered Names for Mandarin Word Identification, In *Proceedings of the COLING 1992*, Page 1239-1243

Liu, Shing-Huan. 1995. An Automatic translator between Traditional Chinese and Simplified Chinese in Unicode. *Proceedings of the 7th International Unicode Conference*, San Jose, California, September 1995

Richard Sproat, Nancy chang, A Stochastic Finite-State Word Segmentation Algorithm For Chinese, 5 May 1994

Packard, Jerome L., editor. 1997, *New Approaches to Chinese Word Formation, Morphology Phonology and the Lexicon in Modern and Ancient Chinese*. Mouton de Gruyter, Berlin, 1997

Thomas Emerson, Segmentating Chinese in Unicode, *Technical report for Basis Technology Corporation*, In *Proceedings of 16th International Unicode Conference*.

Wu, Zimin and Gwyneth Tseng. 1995. ACTS: An automatic Chinese text segmentation system for full text retrieval. *Journal of the American Society for information Science*, 46(2):83 – 96, March 1995

Yuan Chunfa, Huang Changning and Pan Shimei, Knowledge Acquisition and Chinese Parsing Based on Corpus, In *Proceedings of the COLING 1992*, Page 1300-1304

8.4 MITRE reports

David D. Palmer, A Trainable Rule-based Algorithm for Word Segmentation, In *Proceedings for ACL-1997*, 1997

David D. Palmer, Tokenisation and Sentence Segmentation, A Handbook for Natural Language Processing, 1997

David D. Palmer and John D. Burger, Chinese Word Segmentation and Information Retrieval, 1997

John Aberdeen, John Burger, David Day, Lynette Hirschman, Patricia Robinson, and Marc Vilain, Description of the Alembic System Used for MUC-6, Released for publication at 6th Message Understanding Conference (MUC-6), 1997

Lisa Ferro, Marc Vilain, Alexander Yeh, Learning Transformation Rules to Find Grammatical Relations, Released to Computational Natural Language Learning workshop 1999

Marc Villian, Inferential Information Extraction, Internal Report and Guidelines.

8.5 Operating Manuals and ACE/MUC Guidelines

ACE Team Members and Participants, EDT Metonymy Annotation Guidelines Version 2.1, Draft Released at ACE meeting 2001

ACE Team Members and Participants, Entity Detection and Tracking – Phase 1, ACE Pilot Study Task Definition, 2000

Ada Brunstein and Lance Ranshaw, Preliminary Guidelines for Relation Detection and Characterization (RDC), Draft Version 2.2 Released at ACE meeting 2001

Beatrice Santorini, Part-of-Speech Tagging Guidelines for Penn Treebank Project,
University of Pennsylvania, June 1990

Fei Xia, The Segmentation Guidelines for the Penn Chinese Treebank (3.0), University of
Pennsylvania, October 2000

Multilingual Entity Task (MET) Definition for Chinese, 01 March 1996

Shi-Wen Yu, Modern Mandarin Part of Speech Lexicon Guidelines, Beijing University
Institute for Computational Linguistics, 1999

/w /ns /w /ns /w /ns /w /ns /w /ns /v /n /c /n /u /v /n
 /n /c /r /n /m /m /n /v /u /n <LEX> </LEX></S><S> /ns /ns /n
 /v /n /n /f /u /m /q /w /v /a /u /n /n /n /p /a /u /n
 /n <LEX> </LEX></S> <S> /r /n /n /nr /nr /v /w /ns /f /ns /v
 /n /p /ns /n /j /n /v /v /u /m /q
 /b /n /w /t /v <LEX> </LEX></S><S> /r /v /l /ns /v /n /d /a
 /w /v /t /n /a /n /u /n /n /w /t /v
 <LEX> </LEX></S> <S> /r /n /n /nr /nr /v /w /ns /f /ns /f /v /n
 /v /a /u /m /q /v /n /w /t /p /n <LEX> </LEX></S>
 </TEXT>
 </DOC>

Appendix B Selected Code

B1 Sample spec file

```

INCLUDE                                general-specs.spec

*.collection                            CHINESE-LDC
*.language                               ZH

#####
###                                     ###
###   Preprocess                         ###
###                                     ###
#####

prelembic.process?                      NIL

lisp.phases                              "NE"
ne.language=ZH.passes                    "ML-NE"
ne.language=ZH.pass-order                "ML-NE"

##ne.language=ZH.pass=ML-NE.rules        rules/danhu-learned-2-
rules.lisp
ne.language=ZH.pass=ML-NE.rules
    /afs/rcf/project/corporal/chinese/danhu/training_log/rules-exp-44235.lisp

lisp.phase=NE.language=ZH.out-tags       "NR NT NS"
lisp.phase=NE.language=ZH.tag=NR.start-tag "<ENAMEX\ TYPE=\\"PERSON\\">"
lisp.phase=NE.language=ZH.tag=NR.start-tag "<ENAMEX\ TYPE=\\"ORGANIZATION\\">"
lisp.phase=NE.language=ZH.tag=NS.start-tag "<ENAMEX\ TYPE=\\"LOCATION\\">"
lisp.phase=NE.language=ZH.end-tag        "</ENAMEX>"

```

B2 Sample learned rules file for phrasing

The following rules were learned for Organization Tagging

```

;; Rule #1
(def-phraser-rule
  :anchor      :LEXEME
  :conditions  (:NOT (:WD :EVAL #'PART-OF-SOME-PHRASE?))
              (:LWD-1 :LEX :|nt|)
  :actions     (:CREATE :WD :WD :NT)
)

;; Rule #2
(def-phraser-rule
  :conditions  (:PHRASE :PHRASE-LABEL :NT)
              (:LWD-1 :LEX-SUFFIX "»á")
  :actions     (:SET-LABEL :NT)
              (:EXPAND :LEFT-1)
)

;; Rule #3
(def-phraser-rule
  :conditions  (:PHRASE :PHRASE-LABEL :NT)
              (:LEFT-1 :P-O-S :|n|)
  :actions     (:SET-LABEL :NT)
              (:EXPAND :LEFT-1)
)

;; Rule #4
(def-phraser-rule
  :anchor      :LEXEME
  :conditions  (:NOT (:WD :EVAL #'PART-OF-SOME-PHRASE?))
)

```

```

        (:RIGHT-1 :LEX-SUFFIX "Ě¼")
:actions      (:CREATE :WD :WD :NT)
)

;; Rule #5

(def-phraser-rule
 :conditions  (:PHRASE :PHRASE-LABEL :NT)
              (:RIGHT-1 :LEX-SUFFIX "Ě¼")
 :actions    (:SET-LABEL :NT)
              (:EXPAND :RIGHT-1)
)

;; Rule #6

(def-phraser-rule
 :conditions  (:PHRASE :PHRASE-LABEL :NT)
              (:LEFT-1 :LEX :|ns|)
 :actions    (:SET-LABEL :NT)
              (:EXPAND :LEFT-1)
)

;; Rule #7

(def-phraser-rule
 :conditions  (:PHRASE :PHRASE-LABEL :NT)
              (:RWD-1 :LEX "ÖĎňě")
 :actions    (:SET-LABEL :NT)
              (:EXPAND :RIGHT-1)
)

;; Rule #8

(def-phraser-rule
 :conditions  (:PHRASE :PHRASE-LABEL :NT)
              (:LWD-2 :LEX-SUFFIX "Ě¼")
 :actions    (:SET-LABEL :NT)
              (:EXPAND :LEFT-1)
)

;; Rule #9

(def-phraser-rule
 :conditions  (:PHRASE :PHRASE-LABEL :NT)
              (:LEFT-1 :P-O-S :|ns|)
 :actions    (:SET-LABEL :NT)
              (:EXPAND :LEFT-1)
)

;; Rule #10

(def-phraser-rule
 :conditions  (:PHRASE :PHRASE-LABEL :NT)
              (:RWD-1 :LEX :|nr|)
 :actions    (:SET-LABEL :IGNORE)
)

;; Rule #11

(def-phraser-rule
 :conditions  (:PHRASE :PHRASE-LABEL :NT)
              (:LWD-1 :LEX :|j|)
 :actions    (:SET-LABEL :IGNORE)
)

;; Rule #12

(def-phraser-rule
 :anchor      :LEXEME
 :conditions  (:NOT (:WD :EVAL #'PART-OF-SOME-PHRASE?))
              (:RIGHT-1 :LEX "¼-íĀ")
)

```

```

:actions      (:CREATE :WD :WD :NT)
)

;; Rule #13

(def-phraser-rule
:conditions   (:PHRASE :PHRASE-LABEL :NT)
              (:RIGHT-1 :LEX :ZH_ORG_SUFFIX)
:actions      (:SET-LABEL :NT)
              (:EXPAND :RIGHT-1)
)

;; Rule #14

(def-phraser-rule
:conditions   (:PHRASE :PHRASE-LABEL :NT)
              (:LEFT-1 :LEX :|nz|)
:actions      (:SET-LABEL :NT)
              (:EXPAND :LEFT-1)
)

;; Rule #15

(def-phraser-rule
:conditions   (:PHRASE :PHRASE-LABEL :NT)
              (:LEFT-1 :LEX :|ns|)
:actions      (:SET-LABEL :NT)
              (:EXPAND :LEFT-1)
)

;; Rule #16

(def-phraser-rule
:conditions   (:PHRASE :PHRASE-LABEL :NT)
              (:RIGHT-1 :LEX-SUFFIX "%Ö")
:actions      (:SET-LABEL :NT)
              (:EXPAND :RIGHT-1)
)

;; Rule #17

(def-phraser-rule
:conditions   (:PHRASE :PHRASE-LABEL :NT)
              (:LWD-1 :LEX :|v|)
:actions      (:SET-LABEL :NT)
              (:CONTRACT :LWD-2)
)

;; Rule #18

(def-phraser-rule
:conditions   (:PHRASE :PHRASE-LABEL :NT)
              (:RWD-1 :LEX "î-ô±»â")
:actions      (:SET-LABEL :NT)
              (:EXPAND :LEFT-1)
)

;; Rule #19

(def-phraser-rule
:conditions   (:PHRASE :PHRASE-LABEL :NT)
              (:RIGHT-1 :P-O-S :|j|)
:actions      (:SET-LABEL :NT)
              (:EXPAND :RIGHT-1)
)

;; Rule #20

(def-phraser-rule
:anchor       :LEXEME
:conditions   (:NOT (:WD :EVAL #'PART-OF-SOME-PHRASE?))
)

```

```

      (:RIGHT-1 :LEX "ÊÐÎ~")
:actions  (:CREATE :WD :WD :NT)
)

```

B3 Manually Written Rules for Organization Tagging

```

(def-phraser-rule
  :anchor      :lexeme
  :conditions  (:wd :lex-suffix :ZH_ORG_SUFFIX)
              (:wd :p-o-s (:|n| :|j| :|l|))
              (:NOT (:wd :eval #'part-of-some-phrase?))
  :actions    (:create :wd :wd :ORG-HEAD)
  :traced?   NIL)

(def-phraser-rule
  :anchor      :phrase
  :conditions  (:phrase :phrase-label :ORG-HEAD)
              (:wd-any :lex-suffix :ZH_ORG_NEG_SUFFIX)
  :actions    (:set-label :IGNORE))

(def-phraser-rule
  :anchor      :phrase
  :conditions  (:phrase :phrase-label :ORG-HEAD)
              (:wd-any :p-o-s :|n|)
              (:left-1 :p-o-s :|n|)
              (:left-2 :p-o-s :|n|)
              (:left-3 :p-o-s :|ns|)
  :actions    (:expand :left-3)
              (:set-label :ORGANIZATION))

(def-phraser-rule
  :anchor      :phrase
  :conditions  (:phrase :phrase-label :ORG-HEAD)
              (:wd-any :p-o-s :|n|)
              (:left-1 :p-o-s :|n|)
              (:left-2 :p-o-s :|ns|)
              (:left-3 :p-o-s :|ns|)
  :actions    (:expand :left-3)
              (:set-label :ORGANIZATION))

(def-phraser-rule
  :anchor      :phrase
  :conditions  (:phrase :phrase-label :ORG-HEAD)
              (:wd-any :p-o-s :|n|)
              (:left-1 :p-o-s :|vn|)
              (:left-2 :p-o-s :|n|)
              (:left-3 :p-o-s :|n|)
  :actions    (:expand :left-3)
              (:set-label :ORGANIZATION))

(def-phraser-rule
  :anchor      :phrase
  :conditions  (:phrase :phrase-label :ORG-HEAD)
              (:wd-any :p-o-s :|n|)
              (:left-1 :p-o-s :|n|)
              (:left-2 :p-o-s :|nz|)
              (:left-3 :p-o-s :|ns|)
  :actions    (:expand :left-3)
              (:set-label :ORGANIZATION))

(def-phraser-rule
  :anchor      :phrase
  :conditions  (:phrase :phrase-label :ORG-HEAD)
              (:wd-any :p-o-s :|n|)
              (:left-1 :p-o-s (:|ns| :|nz|))
              (:left-2 :p-o-s :|ns|)
  :actions    (:expand :left-2)
              (:set-label :ORGANIZATION))

```

```

(def-phraser-rule
  :anchor      :phrase
  :conditions  (:phrase :phrase-label :ORG-HEAD)
              (:wd-any :p-o-s :|n|)
              (:left-1 :p-o-s :|n|)
              (:left-2 :p-o-s :|ns|)
  :actions    (:expand :left-2)
              (:set-label :ORGANIZATION))

(def-phraser-rule
  :anchor      :phrase
  :conditions  (:phrase :phrase-label :ORG-HEAD)
              (:wd-any :p-o-s :|n|)
              (:left-1 :p-o-s :|nz|)
              (:left-2 :p-o-s :|ns|)
  :actions    (:expand :left-2)
              (:set-label :ORGANIZATION))

(def-phraser-rule
  :anchor      :phrase
  :conditions  (:phrase :phrase-label :ORG-HEAD)
              (:wd-any :p-o-s :|n|)
              (:left-1 :p-o-s (:|vn| :|b|))
              (:left-2 :p-o-s :|ns|)
  :actions    (:expand :left-2)
              (:set-label :ORGANIZATION))

(def-phraser-rule
  :anchor      :phrase
  :conditions  (:phrase :phrase-label :ORG-HEAD)
              (:wd-any :p-o-s :|n|)
              (:left-1 :p-o-s :|n|)
              (:left-2 :p-o-s :|nz|)
  :actions    (:expand :left-2)
              (:set-label :ORGANIZATION))

(def-phraser-rule
  :anchor      :phrase
  :conditions  (:phrase :phrase-label :ORG-HEAD)
              (:wd-any :p-o-s :|n|)
              (:left-1 :p-o-s :|n|)
              (:left-2 :p-o-s :|n|)
              (:left-3 :p-o-s (:|n| :|vn|))
  :actions    (:expand :left-3)
              (:set-label :ORGANIZATION))

(def-phraser-rule
  :anchor      :phrase
  :conditions  (:phrase :phrase-label :ORG-HEAD)
              (:wd-any :p-o-s :|n|)
              (:left-1 :p-o-s :|vn|)
              (:left-2 :p-o-s :|n|)
              (:left-3 :p-o-s :|ns|)
  :actions    (:expand :left-3)
              (:set-label :ORGANIZATION))

(def-phraser-rule
  :anchor      :phrase
  :conditions  (:phrase :phrase-label :ORG-HEAD)
              (:wd-any :p-o-s :|n|)
              (:left-1 :p-o-s :|n|)
              (:left-2 :p-o-s :|n|)
  :actions    (:expand :left-2)
              (:set-label :ORGANIZATION))

(def-phraser-rule
  :anchor      :phrase
  :conditions  (:phrase :phrase-label :ORG-HEAD)
              (:wd-any :p-o-s :|n|)
              (:left-1 :p-o-s (:|n| :|nt|))
  :actions    (:expand :left-1))

```

```

      (:set-label :ORGANIZATION))

(def-phraser-rule
  :anchor      :phrase
  :conditions  (:phrase :phrase-label :ORG-HEAD)
              (:wd-any :p-o-s :|n|)
              (:left-1 :p-o-s (:|nz| :|ns| :|j|))
  :actions    (:expand :left-1)
              (:set-label :ORGANIZATION))

(def-phraser-rule
  :anchor      :phrase
  :conditions  (:phrase :phrase-label :ORG-HEAD)
              (:wd-any :p-o-s :|j|)
              (:left-1 :p-o-s (:|nt| :|ns| :|n|))
  :actions    (:expand :left-1)
              (:set-label :ORGANIZATION))

(def-phraser-rule
  :anchor      :lexeme
  :conditions  (:NOT (:wd :eval #'part-of-some-phrase?))
              (:wd :p-o-s :|nt|)
  :actions    (:create :wd :wd :ORGANIZATION)
  :traced?    NIL)

(def-phraser-rule
  :anchor      :phrase
  :conditions  (:phrase :phrase-label :ORGANIZATION)
              (:left-1 :p-o-s :|ns|)
  :actions    (:expand :left-1))

(def-phraser-rule
  :anchor      :phrase
  :conditions  (:phrase :phrase-label :ORG-HEAD)
  :actions    (:set-label :IGNORE))

```

B4 MMS Segmenter

```
#!/usr/bin/perl
```

```

# Read in the lexicon
open(WRDS, "/afs/rfc/project/corpora1/chinese/danhu/wordlist.txt") or die "Can't open wordlist\n";
while (<WRDS>) {
  chomp;
  next if /^#/;
  $cwords{$_} = 1;
  if (length($_) == 6) {
    if (!exists($cwords{substr($_, 0, 4)})) {
      $cwords{substr($_, 0, 4)} = 2;
    }
  }
  if (length($_) == 8) {
    if (!exists($cwords{substr($_, 0, 4)})) {
      $cwords{substr($_, 0, 4)} = 2;
    }
    if (!exists($cwords{substr($_, 0, 6)})) {
      $cwords{substr($_, 0, 6)} = 2;
    }
  }
  if (length($_) == 10) {
    if (!exists($cwords{substr($_, 0, 4)})) {
7b      $cwords{substr($_, 0, 4)} = 2;
    }
    if (!exists($cwords{substr($_, 0, 6)})) {
      $cwords{substr($_, 0, 6)} = 2;
    }
    if (!exists($cwords{substr($_, 0, 8)})) {
      $cwords{substr($_, 0, 8)} = 2;
    }
  }
}

```

```

    }
  }
}
close(WRDS);

sub addsegword {
  ($_) = shift;
  next if /^#/;
  $cwords{$_} = 1;
  if (length($_) == 6) {
    if (!exists($cwords{substr($_, 0, 4)})) {
      $cwords{substr($_, 0, 4)} = 2;
    }
  }
  if (length($_) == 8) {
    if (!exists($cwords{substr($_, 0, 4)})) {
      $cwords{substr($_, 0, 4)} = 2;
    }
    if (!exists($cwords{substr($_, 0, 6)})) {
      $cwords{substr($_, 0, 6)} = 2;
    }
  }
  if (length($_) == 10) {
    if (!exists($cwords{substr($_, 0, 4)}))
'7b      $cwords{substr($_, 0, 4)} = 2;
    if (!exists($cwords{substr($_, 0, 6)})) {
      $cwords{substr($_, 0, 6)} = 2;
    }
    if (!exists($cwords{substr($_, 0, 8)})) {
      $cwords{substr($_, 0, 8)} = 2;
    }
  }
}

# Numbers
$numbers = "                                ";
$numbers .= "                                ";
for ($n = 0; $n < length($numbers); $n+=2) {
  $numbers{substr($numbers, $n, 2)} = 1;
}

#Classifiers (to follow numbers)
$classifier = "                                ";
$classifier .= "                                ";
for ($n = 0; $n < length($classifier); $n+=2) {
  $classifiers{substr($classifier, $n, 2)} = 1;
}

# Wide ASCII words
$wascii = "                                ";
$wascii .= "                                ";
$wascii .= "                                ";
for ($n = 0; $n < length($wascii); $n+=2) {
  $cascii{substr($wascii, $n, 2)} = 1;
}

# Foreign name transliteration characters
$foreign = "                                ";
$foreign .= "                                ";
$foreign .= "                                ";
$foreign .= "                                ";
$foreign .= "                                ";
for ($n = 0; $n < length($foreign); $n+=2) {
  $cforeign{substr($foreign, $n, 2)} = 1;
}

#Chinese surnames
$surname = "                                ";

```

```

$surname .= "          ";
$surname .= "          ";
$surname .= "          ";
$surname .= "          ";
$surname .= "          ";
$surname .= "          ";
$surname .= "          ";
$surname .= "          ";
$surname .= "          ";
$surname .= "          ";
$surname .= "          ";
$uncommonsurname = "          "; #
for ($n = 0; $n < length($surname); $n+=2) {
    $csurname{substr($surname, $n, 2)} = 1;
}
for ($n = 0; $n < length($uncommonsurname); $n+=2) {
    $uncommoncsurname{substr($uncommonsurname, $n, 2)} = 1;
}

# Add in 2 character surnames; also add to lexicon so they'll be segmented as one unit
$csurname{"  "} = 1; $cwords{"  "} = 1;
$csurname{"  "} = 1; $cwords{"  "} = 1;
$csurname{"  "} = 1; $cwords{"  "} = 1;
$csurname{"  "} = 1; $cwords{"  "} = 1;
$csurname{"  "} = 1; $cwords{"  "} = 1;
$csurname{"  "} = 1; $cwords{"  "} = 1;
$csurname{"  "} = 1; $cwords{"  "} = 1;
$csurname{"  "} = 1; $cwords{"  "} = 1;
$csurname{"  "} = 1; $cwords{"  "} = 1;
$csurname{"  "} = 1; $cwords{"  "} = 1;
$csurname{"  "} = 1; $cwords{"  "} = 1;

$punctuation .= "          . " " ";

#Not in name
$notname = "          ";
$notname .= $punctuation;
for ($n = 0; $n < length($notname); $n+=2) {
    $cnotname{substr($notname, $n, 2)} = 1;
}

sub add_ChineseNames {
    ($tmpline) = @_ ;
    $tlen = length($tmpline);
    $newline = "";
    for ($m = 0; $m < $tlen; $m++) {
        $tchar = substr($tmpline, $m, 1);
        $currtoken = "";
        if ($tchar =~ /\s$/) {
            $newline .= $tchar;
        } else {
            $currtoken = "";
            while ($tchar !~ /\s$/ and $m < $tlen) {
                $currtoken .= $tchar;
                $m++;
                $tchar = substr($tmpline, $m, 1);
            }

            if (defined($csurname{$currtoken}) or
                defined($uncommoncsurname{$currtokenpd})) { # found a surname, see what follows
                # go past following spaces
                $tchar = substr($tmpline, $m, 1);
                $spaces = "";
                while ($tchar =~ /\s/ and $m < $tlen) {
                    $spaces .= $tchar;
                    $m++;
                    $tchar = substr($tmpline, $m, 1);
                }
                # Get next token
            }
        }
    }
}

```

```

    $tchar = substr($tmpline, $m, 1);
    $currtoken2 = "";
    while ($tchar != /\s/ and $m < $tlen) {
        $currtoken2 .= $tchar;
        $m++;
        $tchar = substr($tmpline, $m, 1);
    }
    # go past following spaces
    $tchar = substr($tmpline, $m, 1);
    $spaces2 = "";
    while ($tchar == /\s/ and $m < $tlen) {
        $spaces2 .= $tchar;
        $m++;
        $tchar = substr($tmpline, $m, 1);
    }
    # Get next token
    $tchar = substr($tmpline, $m, 1);
    $currtoken3 = "";
    while ($tchar != /\s/ and $m < $tlen) {
        $currtoken3 .= $tchar;
        $m++;
        $tchar = substr($tmpline, $m, 1);
    }
    if (isChinese($currtoken2) and (length($currtoken2) == 2) and
        (!defined($cnotname{$currtoken2})) and
        isChinese($currtoken3) and length($currtoken3) == 2 and
        !defined($cnotname{$currtoken3}))
    {
        $newline .= $cname[0] . $currtoken . $currtoken2 . $currtoken3 . $cname[1];
        # $words{$currtoken . $currtoken2 . $currtoken3} = 1;
        # $words{$currtoken . $currtoken2} = 2; # short version for checking
    } elseif (isChinese($currtoken2) and (length($currtoken2) == 2)
        and (!defined($cnotname{$currtoken2})))
    {
        $newline .= $currtoken . $currtoken2 . $spaces2 . $currtoken3;
        $words{$currtoken . $currtoken2} = 1;
    } elseif (defined($csurname{$currtoken}) and
        isChinese($currtoken2) and (length($currtoken2) == 4) and
        ($words{$currtoken2} != 1) and
        (!defined($cnotname{$currtoken2})))
    {
        $newline .= $cname[0] . $currtoken . $currtoken2 . $cname[1] . $spaces2 . $currtoken3;
        $words{$currtoken . $currtoken2} = 1;
        $words{$currtoken . substr($currtoken2, 0, 2)} = 2; # short version to check
    } elseif (defined($uncommoncsurname{$currtoken}) and
        isChinese($currtoken2) and (length($currtoken2) == 4)
        and (!defined($cnotname{$currtoken2})))
        and ($words{$currtoken2} != 1))
    {
        $newline .= $cname[0] . $currtoken . $currtoken2 . $cname[1] . $spaces2 . $currtoken3;
        $words{$currtoken . $currtoken2} = 1;
        $words{$currtoken . substr($currtoken2, 0, 2)} = 2; # short version to check
    } else {
        $newline .= $currtoken . $spaces . $currtoken2 . $spaces2 . $currtoken3;
    }
} else {
    $newline .= $currtoken;
}
$m--; # reset so won't skip space
}
}

$newline;
}

#sub cword_start {
#   my($tword) = @_ ;
#   if (grep(/^\$tword/, @cwordlist) > 0) {

```

```

#       return 1;
#   } else {
#       return 0;
#   }
#}

sub isChinese {
    my($cchar) = @_;
    for ($b = 0; $b < length($cchar); $b++) {
        if (unpack("C", substr($cchar, $b, 1)) < 128) {
            return 0;
        }
    }
    return 1;
}

sub allnum {
    ($localnum) = @_;
    # Need this if?
    if ($localnum =~ m/[0-9][0-9]*(\.[0-9]+)?/) {
        return 1;
    }

    for ($k = 0; $k < length($localnum); $k+=2) {
        if (!defined($cnumbers(substr($localnum, $k, 2)))) {
            return 0;
        }
    }
    return 1;
}

sub allnumbers {
    my($localnum) = @_;
    if ($localnum =~ m/[0-9][0-9]*(\.[0-9]+)?/) {
        return 1;
    }

    for ($k = 0; $k < length($localnum); $k+=2) {
        if (!defined($cnumbers(substr($localnum, $k, 2)))) {
            return 0;
        }
    }
    return 1;
}

sub allwascii {
    ($localstr) = @_;
    for ($k = 0; $k < length($localstr); $k+=2) {
        if (!defined($cascii(substr($localstr, $k, 2)))) {
            return 0;
        }
    }
    return 1;
}

sub allforeign {
    ($localstr) = @_;
    for ($k = 0; $k < length($localstr); $k+=2) _7b
        if (!defined($cforeign(substr($localstr, $k, 2)))) {
            return 0;
        }
    }
    return 1;
}

sub segmentline() {
    my($line) = @_;

```

```

$chinaccum = "";
$outline = "";
$linelen = length($line);
for ($i = 0; $i <= $linelen; $i++) {
    $char1 = substr($line, $i, 1);
    if (unpack("C", $char1) > 127) {
        $chinchar = substr($line, $i, 2);
        if ($chinaccum eq "") {
            $outline .= " " unless $i == 0;
            $chinaccum = $chinchar;
        }
        _7d else {
            if (defined($cwords{$chinaccum . $chinchar}) and
                $cwords{$chinaccum . $chinchar} == 1) { # is in lexicon
                $chinaccum .= $chinchar;
            }
            elseif (allnum($chinaccum) and (defined($cnumbers{$chinchar})
                or defined($classifiers{$chinchar}))) {
                $chinaccum .= $chinchar;
            }
            elseif (allwascii($chinaccum) and defined($cascii{$chinchar})) {
                $chinaccum .= $chinchar;
            }
            elseif (allforeign($chinaccum) and defined($cforeign{$chinchar}) and
                defined($cwords{substr($line, $i, 4)}) and
                $cwords{substr($line, $i, 4)} != 1 and
                $cwords{substr($line, $i, 4)} != 2) {
                $chinaccum .= $chinchar;
            }
            elseif (defined($cwords{$chinaccum . $chinchar}) and
                ($cwords{$chinaccum . $chinchar} == 2) and
                defined($cwords{$chinaccum . $chinchar . substr($line, $i+2, 2)}) and
                (($cwords{$chinaccum . $chinchar . substr($line, $i+2, 2)} == 1) or
                ($cwords{$chinaccum . $chinchar . substr($line, $i+2, 2)} == 2)))
                { # starts a word in the lexicon
                $chinaccum .= $chinchar;
            }
            else {
                $outline .= $chinaccum . " ";
                $chinaccum = $chinchar; # start anew
            }
        }
        $i++;
    }
    else { # Plain ascii text, attach any accumulated Chinese and then ascii
        if ($chinaccum ne "") _7b
            $outline .= $chinaccum . " ";
            $chinaccum = "";
        }
        $outline .= $char1;
    }
}

$chinline = add_ChineseNames($outline);
$chinline;
}

```

B5 Translating wordlists from Web based dictionary

```

#!/afs/rcf/lang/bin/perl

srand;

#Build arrays of the words: [english word][type][chinese word]
open(INFILE, "< /afs/rcf/project/empty/data/dictionary/chinese/online/ntw.types") or die
"can't open file";
$i=0;
while(<INFILE>){
    chomp;
    if ($_ !~ m/^\;/){
        $list1{$_} = rand;
        $list2[$i] = $_;
        $i++;
    }
}

```

```

@list3 = sort{$list1{$a} <=> $list1{$b}} @list2;

foreach $enword (@list3){
    print $enword."\n";
}

#Prompt ciba-online for the Chinese word

sub pipecmd
{
    my $cmd = shift(@_);
    if (open(SENDOUT,"|$cmd")) {
        foreach $cmd (@_) {
            print SENDOUT $cmd."\n";
        }
        close SENDOUT;
    } else {
        die;
    }
}

#exit(0);
open(OUT, "> /afs/rcf/project/corporal/chinese/danhu/zh-geo-world-provinces.txt") or die
"can't open output file";

foreach $enword (@list3){
    $enword =~ s/^\s+$//;
    $wordlen = length($enword)+6;

    pipecmd (
        "/afs/rcf/user/jhndrsn/bin/nc webproxyl.mitre.org 80 > tempword.html",
        "POST http://ciba.kingsoft.net/cgi-bin/ecdetail.cgi HTTP/1.0 ",
        "Referer: http://ciba.kingsoft.net/online/main.html",
        "User-Agent: Mozilla/4.75 [en] (X11; U; SunOS 5.7 sun4u)",
        "Host: ciba.kingsoft.net:80",
        "Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png,
*/*",
        "Accept-Encoding: gzip",
        "Accept-Language: en-US, en, en-GB, de, zh-CN, zh-TW, cs",
        "Accept-Charset: iso-8859-1,*utf-8",
        "Content-type: application/x-www-form-urlencoded",
        "Content-length: $wordlen",
        "",
        "terms=$enword",
    );
    $newhtmlfile =
"/afs/rcf/project/corporal/chinese/danhu/ntwhtmlfiles/.$enword.".05082001.html";
    print $newhtmlfile."\n";
    pipecmd (
        "cp tempword.html $newhtmlfile",
    );

    exit(0);
    print $enword."\n";
    undef($/);
    open(HTML, "< tempword.html") or die "unable to find tempword.html";
    $htmltext = <HTML>;
    close(HTML);
    $chword = extractmeaning($htmltext);
    $wordhash{$enword} = ["$chword", "countries"];
    print $enword."\n";
    sleep 1;
}

#Extract the Chinese word from the junk that is passed back and place it in the array

sub extractmeaning{
    my($mytext) = @_;
    undef($/);
    $mytext =~ s/^\.*\[\í"óã\]//s;

```

```

if($mytext =~ m/\(*¶ÔÓ|´ÊÏò²»´æÔÛ\)/){
  #print "(*¶ÔÓ|´ÊÏò²»´æÔÛ)\n";
  return 0;
}else{
  $mytext =~ s/^\.*n\.//s;
  $i=0;
  $mychword = "";
  while((substr($mytext, $i, 1) ne ",") and
        (substr($mytext, $i, 1) ne ")") and
        (substr($mytext, $i, 1) ne "<")){
    $mychword .= substr($mytext, $i, 1);
    #print "$mychword\n";
    $i++;
    #print substr($mytext, $i, 1)." \n";
  }
  $mychword =~ s/^\s+//;
  #print "$mychword\n";
  return $mychword;
  #print $mytext;
}

}

#Print the resulting array however deemed most useful

foreach $enword (%wordhash){
  @array = @{$wordhash{$enword}};
  print $array[0]." ".$array[1]."\n";
  printf( "%15s%15s%20s\n", $enword, $array[0], $array[1]);
}

```

Appendix C Sample Score File

C1 Evaluation output of the trained phraser

```

;;; Parameters presumably in effect when rules were derived:
;;; Host machine on which learner executed (excl::*current-site-name*) = delphi.mitre.org
;;;
;;; *RELEVANT-LOCS* =
;;;     LEFT-2 LEFT-1 WD-ANY WD-SPAN
;;;     RIGHT-1 RIGHT-2 LWD-1 LWD-2
;;;     RWD-1 RWD-2
;;; *EXTERNAL-LOCS* =
;;;     LEFT-1 LEFT-2 RIGHT-1 RIGHT-2
;;;     RIGHT-3 LEFT-3
;;; *COMPLEX-LOC-RESTRICTIONS* =
;;;     (2 RIGHT-1 RIGHT-2) (2 RIGHT-2 RIGHT-1) (2 LEFT-1 LEFT-2) (2 LEFT-2 LEFT-1)
;;;     (2 LEFT-1 RIGHT-1) (2 RIGHT-1 LEFT-1) (2 LWD-1 LEFT-1) (2 RWD-1 RIGHT-1)
;;; *COMPLEX-LOC-INCLUSION-PCTS* =
;;;     (1 1.0) (2 0.5) (3 0.2) (4 0.1)
;;;     (5 0.1) (6 0.1)
;;; *RELEVANT-KEYS* =
;;;     NT-CONTRACT-RIGHT-1 NT-CONTRACT-LEFT-1 NT-EXPAND-LEFT-1 NT-EXPAND-RIGHT-1
;;;     NT-SEED NT
;;; *IRRELEVANT-ANSWERS* =
;;; *ACTIVE-RULE-PREDICATES* =
;;;     VB-CLASS? NN-CLASS? NNP-CLASS? ADV-CLASS?
;;;     JJ-CLASS? PART-OF-GEO-PHRASE? PART-OF-HUMAN-PHRASE? PART-OF-ORG-PHRASE?
;;;     PART-OF-CORP-PHRASE? ELSEWHERE-AS-ORG? ELSEWHERE-AS-GEO? ELSEWHERE-AS-HUMAN?
;;;     ELSEWHERE-AS-CORP? PERSON-WORD? CITIZENSHIP-WORD? DATE-WORD?
;;;     CARDINAL-WORD? STATE-WORD? GEO-CONTINENT-WORD? GEO-THE-WORD?
;;;     GEO-MAJOR? GEO-ABBREV? COUNTRY-WORD? ORG-WORD?
;;;     TITLE-WORD? GAZ-WORD? SPOKESPERSON-WORD? SUBSIDIARY-WORD?
;;;     CORP-WORD?
;;;
;;; *TRAINING-REGIME-NAME* = NIL
;;; *INTERNAL-LOCS* = (:LWD-1 :LWD-2 :RWD-1 :RWD-2)
;;; *RELEVANT-OBJ-TYPES* = (:LEX :POS :PREDICATE :WORD-LIST)
;;; *ALIGNMENT-LOCS* = (:LEFT-1 :LEFT-2 :RIGHT-1 :RIGHT-2)
;;; *SCORING-FUNCTION* = :F-MEASURE
;;; *REFINEMENT-SCORING-FUNCTION* = :YIELD-MINUS-SACRIFICE
;;; *F-MEASURE-BETA* = 0.8
;;; *MAX-RULE-CONDITIONALITY* = 1
;;; *USE-RULE-PREDICATE-INVERSES?* = NIL
;;; *SET-COMPLEX-LOC-RESTRICTIONS-RANDOMLY?* = NIL
;;; *SIMPLE-LOC-RESTRICTIONS* = :*
;;; *PHRASER-LANGUAGE* = NIL
;;; *IRRELEVANT-KEYS* = NIL
;;; *RELEVANT-ANSWERS* = (:NT-CONTRACT-RIGHT-1 :NT-CONTRACT-LEFT-1 :NT-EXPAND-LEFT-1
;;;     :NT-EXPAND-RIGHT-1 :NT-SEED :NT)
;;; *MISALIGNMENT-RESOLUTION* = 3
;;; *ALIGNMENT-YIELD-OUTLIERS-FACTOR* = 2
;;; *TRAINING-SET-NAME* = "Unspecified"
;;; *UNIVERSE-SIZE* = 300
;;; *INITIAL-PHRASING-TYPE* = :NO-PHRASING
;;; *MAX-RULE-CONDITIONALITY* = 1
;;; *INITIAL-TEST-SENT* = NIL
;;; *UNIVERSE-SIZE* = 300
;;; *TESTING-SIZE* = 100
;;; *TRAINING-SIZE* = 200
;;; *CACHE-RULE-PREDICATES?* = T
;;; *RESOLVE-MISALIGNMENTS?* = T
;;; *RESOLVE-MISLABELINGS?* = T
;;; *RESOLVE-INITIAL-PHRASING?* = T
;;; *REJECT-REPEATED-RULES?* = 6
;;; *AFFIX-LENGTHS* = (2)
;;; *GENERATE-IGNORE-RULES?* = T
;;; *NORMALIZE-ALL-STRINGS?* = T
;;; *GENERATE-NUMERIC-COMPLEX-OBJECTS?* = NIL
;;; *ALWAYS-PURSUE-REFINEMENT* = T

```

```

;;; *ENABLE-GROWING-IN-EXPAND-PHRASE* = NIL
;;; *PHRASES-FROM-PREPROCESSING* = NIL
;;; *GENERATE-IGNORE-RULES?* = T
;;; *EPOCH-DEPENDENT-CODE* = NIL
;;; *RANDOM-TIE-ARBITRATION?* = 50
;;; *RANDOM-LEXEME-SCORING-RATE* = NIL
;;; *RANDOM-SENTENCE-SELECTION-RATE* = 50
;;; *ACTIVE-RULE-PREDICATES-SOURCE* = *ACTIVE-RULE-PREDICATES-FOR-NE*
;;; *ORIGINAL-TESTING-SENTS-SOURCE* = "Unspecified"
;;; *ORIGINAL-TRAINING-SENTS-SOURCE* = "Unspecified"
;;;(length *all-sentences*) = 0
;;;(length *training-sentences*) = 16227
;;;(length *original-testing-sents*) = 2641
;;;(length *original-training-sents*) = 16227

```

```
;; Data generated on Thu 3-May-01 at 12:05:23 PM
```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Applying learned rules to TRAINING SET sentences. ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

No.	Key	Ans	Corr	Miss	Sprr	R(%) [Lbl]	P(%) [Lbl]	F-measur	Fired
0	3665	0	0	3665	0	0.0	--	--	0
1	3665	3080	1010	2655	2070	27.557981	32.792206	30.53	3080
2	3665	3033	1065	2600	1968	29.058662	35.11375	32.47	372
3	3665	3007	1134	2531	1873	30.941338	37.712006	34.74	346
4	3665	3597	1134	2531	2463	30.941338	31.52627	31.30	590
5	3665	3591	1225	2440	2366	33.424286	34.11306	33.84	597
6	3665	3591	1422	2243	2169	38.799458	39.599	39.28	319
7	3665	3583	1507	2158	2076	41.11869	42.059727	41.69	245
8	3665	3576	1478	2187	2098	40.32742	41.331097	40.93	502
9	3665	3575	1556	2109	2019	42.45566	43.524475	43.10	104
10	3665	3180	1555	2110	1625	42.42838	48.899372	46.15	395
11	3665	2837	1536	2129	1301	41.90996	54.1417	48.61	343
12	3665	2977	1536	2129	1441	41.90996	51.59557	47.33	140
13	3665	2966	1582	2083	1384	43.165077	53.33783	48.85	170
14	3665	2962	1605	2060	1357	43.792633	54.186363	49.59	55
15	3665	2962	1659	2006	1303	45.26603	56.009453	51.26	66
16	3665	2962	1705	1960	1257	46.521145	57.562458	52.68	47
17	3665	2962	1745	1920	1217	47.612553	58.912895	53.92	178
18	3665	2952	1742	1923	1210	47.530697	59.01084	53.93	118
19	3665	2952	1764	1901	1188	48.13097	59.7561	54.61	34
20	3665	3018	1764	1901	1254	48.13097	58.449303	53.94	66

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Testing learned rules against independent test set of sentences. ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

No.	Key	Ans	Corr	Miss	Sprr	R(%) [Lbl]	P(%) [Lbl]	F-measur	Fired
0	728	0	0	728	0	0.0	--	--	0
1	728	523	168	560	355	23.076923	32.12237	27.86	523
2	728	517	171	557	346	23.48901	33.075436	28.53	53
3	728	510	180	548	330	24.725275	35.294117	30.25	52
4	728	608	180	548	428	24.725275	29.605263	27.49	98
5	728	608	203	525	405	27.884615	33.388157	31.00	109
6	728	608	231	497	377	31.730768	37.993423	35.28	57
7	728	608	245	483	363	33.653847	40.296055	37.41	46
8	728	608	231	497	377	31.730768	37.993423	35.28	99
9	728	608	246	482	362	33.791206	40.460526	37.57	16
10	728	554	246	482	308	33.791206	44.40433	39.56	54
11	728	523	246	482	277	33.791206	47.036327	40.80	31
12	728	542	246	482	296	33.791206	45.38745	40.03	19
13	728	540	253	475	287	34.752747	46.851852	41.25	20
14	728	540	254	474	286	34.89011	47.037037	41.41	2
15	728	540	257	471	283	35.302197	47.59259	41.90	4
16	728	540	260	468	280	35.714287	48.148148	42.39	6
17	728	540	265	463	275	36.4011	49.074074	43.20	40
18	728	539	263	465	276	36.126373	48.794064	42.92	25
19	728	539	268	460	271	36.813187	49.721706	43.74	6

20 | 728 549 268 460 281 | 36.813187 48.81603 43.31 | 10

C2 Evaluation output of the manual phraser

* * * SUMMARY SCORES * * *

	POS	ACT	COR	PAR	INC	MIS	SPU	NON	REC	PRE	UND	OVG	SUB	ERR
SUBTASK SCORES														
enamex														
organizatio	801	934	684	0	0	117	250	0	85	73	15	27	0	35
person	0	0	0	0	0	0	0	0	0	0	0	0	0	0
location	0	0	0	0	0	0	0	0	0	0	0	0	0	0
other	0	0	0	0	0	0	0	0	0	0	0	0	0	0
timex														
date	0	0	0	0	0	0	0	0	0	0	0	0	0	0
time	0	0	0	0	0	0	0	0	0	0	0	0	0	0
other	0	0	0	0	0	0	0	0	0	0	0	0	0	0
numex														
money	0	0	0	0	0	0	0	0	0	0	0	0	0	0
percent	0	0	0	0	0	0	0	0	0	0	0	0	0	0
other	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SECT SCORES														
doc	1602	1868	1292	0	76	234	500	0	81	69	15	27	6	39
OBJ SCORES														
enamex	801	934	684	0	0	117	250	0	85	73	15	27	0	35
numex	0	0	0	0	0	0	0	0	0	0	0	0	0	0
timex	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SLOT SCORES														
enamex														
text	801	934	608	0	76	117	250	0	76	65	15	27	11	42
type	801	934	684	0	0	117	250	0	85	73	15	27	0	35
status	0	0	0	0	0	0	0	0	0	0	0	0	0	0
alt	0	0	0	0	0	0	0	0	0	0	0	0	0	0
numex														
text	0	0	0	0	0	0	0	0	0	0	0	0	0	0
type	0	0	0	0	0	0	0	0	0	0	0	0	0	0
status	0	0	0	0	0	0	0	0	0	0	0	0	0	0
alt	0	0	0	0	0	0	0	0	0	0	0	0	0	0
timex														
text	0	0	0	0	0	0	0	0	0	0	0	0	0	0
type	0	0	0	0	0	0	0	0	0	0	0	0	0	0
status	0	0	0	0	0	0	0	0	0	0	0	0	0	0
alt	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ALL SLOTS	1602	1868	1292	0	76	234	500	0	81	69	15	27	6	39
F-MEASURES									P&R	2P&R			P&2R	
									74.47	71.19			78.06	

Appendix D Supplemental Wordlists for Phraser

D1 Surname list

--	--	--	--	--	--	--	--

D2 Foreign phonetic translation characters

--	--	--	--	--	--	--	--

D3 Organization and Location Suffixes

Org	Org	Org	Loc	Loc	Loc

Appendix E Sample Training Data and Other Pre-tagged Corpus

E1 University of Pennsylvania Chinese Treebank project

```

<DOC>
<DOCID> CB302002.BFY 1094 </DOCID>
<HEADER>
<DATE> 1994-09-02 09:31:23 3B:spc011 </DATE>
</HEADER>
<BODY>
<HEADLINE>
((IP-HLN (NP-SBJ (NP-PN (NR )) (NP (NN ) (NN ) (NN ))) (VP (VV )))
</HEADLINE>
<HEADLINE>
((FRAG (NR ) (NR ) (NT ) (NT ) (NN ) (NN ) (PU ) (NR ) (NN ) (NN ) (NN ) (VV
)))
</HEADLINE>
<HEADLINE>
((FRAG (NR ) (NN ) (NR )))
</HEADLINE>
<TEDNPT>
<P>
<S>
((IP (NP-SBJ (NP-PN (NR )) (NP (NN ) (NN ) (NN ))) (VP (VP (VV ) (AS ) (NP-OBJ (NP-APP (NN
)) (PU ) (NN ) (PU ) (NN ))) (QP (CD ) (CLP (M ))) (NP (NN ))) (PU ) (VP (NP-TMP (NT )) (ADVP
(AD )) (VP (VV ))) (PU )) </S> <S> ((IP (NP-SBJ (DNP (NP-PN (QP (OD ) (CLP (M ))) (NP (NP-PN (NR
)) (CC ) (NP (NP (NN )) (NP-PN (NR )) (NP (NN ))) (NP (NN ) (NN ))) (DEG )) (NP (NN )))
(PU ) (VP (ADVP (AD )) (ADVP (AD )) (VP (PP-BNF (P ) (NP (PN ))) (VP (VV ) (NP-OBJ (CP (WHNP-1 (-
NONE- *OP*)) (CP (IP (NP-SBJ (-NONE- *T*-1)) (VP (VA ))) (DEC ))) (NP (NN ))))) (PU )) </S>
</P>
<P>
<S>
((IP (DP-SBJ (DNP (NP (NN ) (NN ) (NN ) (NN )) (DEG )) (DP (DT ) (QP (CD ) (CLP (M ))))) (VP
(ADVP (AD )) (VP (VV ) (NP-OBJ (DNP (NP (NN ) (CC ) (NN )) (DEG )) (NP (NN ) (NN ))))) (PU ))
) </S>
</P>
<P>
<S>
((IP (LCP-TMP (IP (NP-PN-SBJ (NR ) (NN ) (NN ))) (VP (VV ))) (LC )) (PU ) (NP-SBJ (DP (DT ))
(NP (NN ))) (VP (ADVP (AD )) (VP (VV ) (NP-OBJ (NP (NN ) (NN ) (NN )) (CC ) (NP (NP (NN )
(PU ) (NN )) (NP (NN ))))) (PU )) </S>
<S>
((IP (IP (NP-SBJ (NN ) (NN )) (VP (VV ) (NP-OBJ (NP-APP (NN ) (NN ) (ETC )) (NP (NN ))))) (PU
) (IP (NP-SBJ (NN )) (VP (VV ) (NP-OBJ (NN )))) (PU )) </S>
<S>
((IP (PP (P ) (NP (DNP (NP (NN ) (NN )) (DEG )) (QP (CD )) (NP (NN )))) (PU ) (NP-SBJ (NN )
(NN ) (NN )) (VP (VV ) (VP (VV )) (PU )) </S>
</P>
<P>
<S>
((IP (NP-SBJ (NN )) (VP (DP-TMP (DT ) (CLP (M ))) (ADVP (AD )) (VP (VV ) (NP-OBJ (NN )) (IP (NP-SBJ (-
NONE- *PRO*)) (VP (VV ) (PP (P ) (IP (NP-SBJ (-NONE- *pro*)) (VP (VV ) (NP-OBJ (NN ) (NN ) (NN
)))))) (PU ))))
</S>
<S>
((IP (NP-SBJ (DNP (ADJP (JJ )) (DEG )) (NP (NN ) (NN ) (NN ))) (VP (PP-TMP (P ) (NP (NT )
(NN ))) (VP (VV ) (NP-OBJ (NN )))) (PU ))
</S>
<S>
((IP (NP-SBJ (NN )) (VP (VP (VV ) (NP-OBJ (NN )))) (PU ) (VP (VV ) (NP-OBJ (NN )))) (PU )) </S>
<S>
((IP (NP-SBJ (NN ) (NN ) (NN )) (VP (ADVP (AD )) (VP (VV )) (PU )) </S>
</P>
</TEDNPT>
</BODY>
</DOC>

```

E2 Beijing University Institute of Computational Linguistics People's Daily project

19980101-01-001-001/m /v /v /n /u /a /n /w /t /t /n /w /v /n /m /q
/w

19980101-01-001-002/m /nt /n /w /n /n /nr /nr

19980101-01-001-003/m /w /t /t /w /nt /n /w /n /n /nr /nr /v /t /t

19980101-01-001-004/m /n /w /v /v /n /u /a /n /w /w /w /nt /n /nr /nr /Vg /w

19980101-01-001-005/m /n /k /w /n /k /w /n /k /w /n /k /w

19980101-01-001-006/m /p /t /v /f /w /r /m /a /u /p [/n /n /vn /n]nt
/w [/ns /n /vn /n]nt /c [/n /n]nt /w /p /n /r /n /w /p [/ns /a
/n]ns /n /w /ns /c /ns /n /w /s /n /w /p /n /r /u /n /k /w /v
/a /u /vn /c /a /u /vn /w

19980101-01-001-007/m /t /w /v /ns /vn /n /f /d /a /u /d /d /a /u /m /q
/w /ns /n /d /v /nr /nr /n /u /n /w /v /p /v /v /ns /n /n /n
/v /v /w [/ns /n]nt /ad /v /p /ns /v /n /w /c /p "/w /j "/w /w
/l "/w /w /d /v /u /n /v /ns /u /an /an /w [/ns /n]nt /a /u /v
/u /m /q /n /n /w /v /n /a /n /w /v /m /n /w /v /a /u
/n /w /v /u /ns /v /n /v /u /vn /n /w

19980101-01-001-008/m /p /r /m /q /f /w /ns /u /vn /vn /c /vn /vn /v /v
/v /w /n /v /u "/w /a /vn /w /a /j "/w /u /a /vn /n /w /n /n /d
/v /a /u /n /w /n /vn /v /v /w /n /vn /d /v /w /vn /n /n /n
/c /vn /d /v /w /a /n /vn /w /n /vn /c /r /r /n /d /v /a /u
/vn /w /r /m /v /t /m /n /m /n /c /n /v /u /n /n /w /r /v
/p /r /n /c /n /u /an /c /v /u /n /vn /w /n /v /d /v /vn /w
/c /w /ns /v /c /v /u /n /v /v /u /an /w

19980101-01-001-009/m /p /r /m /q /f /w /ns /u /n /vn /v /u /a /n /w /p
/n /v /w /ns /p /ns /w /ns /w /ns /w /ns /u /n /v /n /n /n /t
/v /u /n /c /vn /n /w /ns /p /n /n /c /b /l /u /a /vn /d /v
/w /ns /ad /v [/j /j /j /n]nt /u /vn /w /v /u /ns /w /j /j /c /ns /w
/ns /n /b /vn /w /r /n /vn /w /v /n /c /v /u /n /n /w /v /n
/v /v /u /n /w /p /v /n /n /u /a /vn /c /b /vn /v /u /a /u
/n /w

19980101-01-001-010/m /t /w /ns /n /d /l /u /v /a /u /n /w /c /r /p
/n /n /v /f /d /v /m /an /w /c /r /v /n /u /vn /w /v /v /v /a
/m /q /f /v /u /a /n /c /v /u /a /n /w /d /v /r /u /r /a /n /w
/r /d /v /v /r /an /w /v /l /w /c /r /d /i /w /i /w /v
/n /w /l /w /v /v /ns /n /n /u /n /c /v /d /v /a /w

19980101-01-001-011/m /v /n /u /a /vn /w /v /s /n /ns /n /u /b /n /w
/p /j /j /n /u /vn /c /an /w /p "/w /j "/w /n /c /ns /w /n /w /w
/t /t /ns /u /vn /d /v /ad /v /w

19980101-01-001-012/m /ns /v /ns /n /l /u /m /n /w /v /n /vn /w /v
/i /w /n /w /l /w /r /v /v "/w /m /q /ns "/w /w "/w /j "/w /w "/w /ns /v
"/w /u /n /w /d /v /v /v /w /v /ns /n /p /n /n /v /w /v /v /n /w
/v /a /u /vn /w /v /n /n /n /vn /c /n /vn /w /v /n /ad /v /w
/v /w /v /u /d /v /w /c /d /v /r /v /u /p /m /ns /u /n /f /n
/v /vn /u /a /vn /w

19980101-01-001-013/m /v /n /w /d /a /u /n /n /vn /w /i /u /n /vn /w
/d /p /r /n /u /vn /v /n /n /w /c /w /n /d /d /a /w /f /f /u
/n /n /v /v /w /n /vn /l /w /d /a /d /a /u /a /u /n /n /n /n
/d /v /a /vn /w /l /p /a /u /n /n /vn /f /d /v /n /n /w /n
/u /vn /c /vn /d /v /q /vn /c /vn /w /n /c /vn /u /n /v /a /u /w
/m /n /d /v /d /n /u /n /w /c /v /u /n /d /v /d /d /v /i /w /n
/v /n /r /n /v /d /a /v /w /d /v /d /v /a /a /u /n /n /n /a
/n /w

19980101-01-001-014/m [/ns /n]nt /d /v /v /v /i /u /n /n /n /w /p /l
/m /q /n /u /n /f /ad /v /p /n /r /u /a /n /w /ns /v /v /p /nt
/c /r /n /n /u /vn /w /v /p /v /j /n /vn /w /v /n /w /v /an /w
/v /n /vn /u /n /u /n /vn /w /ns /d /v /v /n /n /c /an /u /a
/n /w /ns /n /v /p /n /r /n /d /w /p /v /a /n /w /d /v /u /a
/n /c /l /w

19980101-01-001-015/m /p /r /l /u /a /n /w /r /v /r /t /a /w /n /a /w

19980101-01-001-016/m /v /w /w /nt /ns /t /t /n /w

E3 MET 2 and BBN data

```

<DOC>
<METID> 001 </METID>
<ID> raw/950715/tg/gnxw0718.15 </ID>
<DATE> 1995 07 18 </DATE>
<HL> </HL>
<AU> Domestic News Dept. </AU>
<TEXT>
<p>
    <ENAMEX TYPE="PERSON"> </ENAMEX> <TIMEX TYPE="DATE"> </TIMEX><ENAMEX
TYPE="LOCATION"> </ENAMEX> <ENAMEX TYPE="LOCATION"> </ENAMEX> <ENAMEX
TYPE="DATE"> </TIMEX> <ENAMEX TYPE="LOCATION"> </ENAMEX><ENAMEX
TYPE="LOCATION"> </ENAMEX> <ENAMEX TYPE="LOCATION"> </ENAMEX> <ENAMEX
TYPE="LOCATION"> </ENAMEX> <ENAMEX TYPE="LOCATION"> </ENAMEX> <ENAMEX
TYPE="LOCATION"> </ENAMEX> <ENAMEX TYPE="LOCATION"> </ENAMEX> <ENAMEX
TYPE="LOCATION"> </ENAMEX> <ENAMEX TYPE="LOCATION"> </ENAMEX> <ENAMEX
TYPE="LOCATION"> </ENAMEX> <ENAMEX TYPE="LOCATION"> </ENAMEX> <ENAMEX
TYPE="PERSON"> </ENAMEX> <ENAMEX TYPE="LOCATION"> </ENAMEX> <ENAMEX
TYPE="ORGANIZATION"> </ENAMEX> <ENAMEX TYPE="ORGANIZATION"> </ENAMEX>
<TIMEX TYPE="DATE"> </TIMEX> <ENAMEX
TYPE="LOCATION"> </ENAMEX> <ENAMEX
TYPE="PERSON"> </ENAMEX> <ENAMEX TYPE="LOCATION"> </ENAMEX> <ENAMEX
TYPE="ORGANIZATION"> </ENAMEX> <TIMEX TYPE="DATE">
</TIMEX> <ENAMEX
TYPE="LOCATION"> </ENAMEX> <ENAMEX
TYPE="PERSON"> </ENAMEX> <ENAMEX TYPE="LOCATION"> </ENAMEX> <TIMEX
TYPE="DATE"> </TIMEX> <ENAMEX TYPE="LOCATION"> </ENAMEX>
<ENAMEX TYPE="LOCATION"> </ENAMEX> <ENAMEX
TYPE="ORGANIZATION"> </ENAMEX> <ENAMEX TYPE="DATE"> </TIMEX> <ENAMEX
TYPE="LOCATION"> </ENAMEX> <TIMEX
TYPE="DATE"> </TIMEX> <NUMEX
X TYPE="MONEY"> </NUMEX> <ENAMEX TYPE="ORGANIZATION">
</ENAMEX> <ENAMEX TYPE="ORGANIZATION"> </ENAMEX> <ENAMEX
TYPE="LOCATION"> </ENAMEX><ENAMEX TYPE="LOCATION"> </ENAMEX> <ENAMEX
TYPE="LOCATION"> </ENAMEX> <ENAMEX TYPE="ORGANIZATION">
</ENAMEX> <ENAMEX TYPE="ORGANIZATION"> </ENAMEX> <ENAMEX
TYPE="ORGANIZATION"> </ENAMEX> <ENAMEX TYPE="ORGANIZATION"> </ENAMEX> <ENAMEX
TYPE="ORGANIZATION"> </ENAMEX> <ENAMEX
TYPE="LOCATION"> </ENAMEX> <TIMEX TYPE="DATE"> </TIMEX>
<NUMEX TYPE="MONEY"> </NUMEX> <ENAMEX
TYPE="ORGANIZATION"> </ENAMEX> <ENAMEX TYPE="LOCATION"> </ENAMEX><ENAMEX
TYPE="LOCATION"> </ENAMEX><ENAMEX TYPE="LOCATION"> </ENAMEX> <ENAMEX
TYPE="LOCATION"> </ENAMEX>
<TIMEX TYPE="DATE"> </TIMEX>
<ENAMEX
TYPE="ORGANIZATION"> </ENAMEX> <ENAMEX
TYPE="LOCATION"> </ENAMEX> <ENAMEX
TYPE="LOCATION"> </ENAMEX> <TIMEX TYPE="DATE"> </TIMEX><ENAMEX
TYPE="LOCATION"> </ENAMEX> <ENAMEX TYPE="LOCATION"> </ENAMEX>
. <TIMEX TYPE="DATE"> </TIMEX> <ENAMEX TYPE="LOCATION">
</ENAMEX> <NUMEX TYPE="MONEY">
</NUMEX> <ENAMEX TYPE="ORGANIZATION"> </ENAMEX> <ENAMEX TYPE="LOCATION">
</ENAMEX><ENAMEX TYPE="LOCATION"> </ENAMEX> <ENAMEX
TYPE="LOCATION"> </ENAMEX> <ENAMEX TYPE="LOCATION">
</ENAMEX> <ENAMEX TYPE="LOCATION"> </ENAMEX> <ENAMEX TYPE="LOCATION">
</ENAMEX> <ENAMEX TYPE="LOCATION"> </ENAMEX> <ENAMEX
TYPE="LOCATION"> </ENAMEX> <ENAMEX
TYPE="LOCATION"> </ENAMEX> <ENAMEX TYPE="ORGANIZATION"> </ENAMEX> <ENAMEX
TYPE="LOCATION"> </ENAMEX> <ENAMEX
TYPE="ORGANIZATION"> </ENAMEX>
<ENAMEX> <ENAMEX TYPE="LOCATION">
</NUMEX TYPE="PERCENT">
</NUMEX> <ENAMEX TYPE="LOCATION">
</ENAMEX> <ENAMEX TYPE="LOCATION"> </ENAMEX>
<NUMEX TYPE="PERCENT"> </NUMEX>

```

<p>

```
<ENAMEX TYPE="PERSON"> </ENAMEX> <ENAMEX TYPE="PERSON">
</ENAMEX> /
</TEXT>
</DOC>
```

E4 Frequency data from PKU People's Daily project

There are 337724 words of the length 1.
 There are 492879 words of the length 2.
 There are 51116 words of the length 3.
 There are 19375 words of the length 4.
 There are 5059 words of the length 5.

POS	Total	1 Char	2 Char	3 Char	4 Char	Other
Ag	321	321	0	0	0	0
na	1	0	0	1	0	0
Mg	7	7	0	0	0	0
vv	1	0	1	0	0	0
vd	528	17	501	10	0	0
nr	35262	18537	13927	1415	623	760
Yg	1	1	0	0	0	0
a	34473	12834	21577	54	8	0
ns	27890	0	18584	7297	1354	655
nt	3573	0	7	3118	416	32
b	8715	1913	6120	646	26	10
c	25479	18923	6228	176	152	0
d	47811	29845	16846	1117	3	0
vn	42734	127	41747	857	3	0
e	27	23	4	0	0	0
ad	5933	496	5435	0	2	0
nx	452	112	49	133	50	108
f	17201	12627	4499	75	0	0
h	47	47	0	0	0	0
nz	3700	0	2090	1034	437	139
i	5033	0	0	8	4966	59
Bg	8	8	0	0	0	0
j	10323	3723	3896	2169	518	17
k	928	928	0	0	0	0
Dg	135	135	0	0	0	0
l	6015	0	26	686	5104	199
n	236810	16846	186882	26800	5258	1024
an	2837	68	2769	0	0	0
o	70	9	51	1	9	0
p	39925	36339	3585	1	0	0
q	24244	22048	1909	235	40	12
r	32327	13800	18395	132	0	0
s	3836	0	3655	179	2	0
t	20680	0	14019	3579	177	2905
Ng	4562	4562	0	0	0	0
u	74829	74363	466	0	0	0
v	184764	64953	118616	1176	17	2
y	1889	1857	32	0	0	0
Rg	10	10	0	0	0	0
z	1391	0	963	217	210	1
Tg	496	496	0	0	0	0
Vg	1749	1749	0	0	0	0

E5 Frequency data from U. of Penn. Treebank project

There are 27187 words of the length 1.
 There are 47216 words of the length 2.
 There are 6330 words of the length 3.
 There are 857 words of the length 4.
 There are 468 words of the length 5.

POS	Total	1 Char	2 Char	3 Char	4 Char	Other
NR	7664	558	4555	2077	314	160
NT	2651	14	1783	378	121	355
SP	17	15	2	0	0	0
BA	112	112	0	0	0	0
AD	4980	2668	2120	182	10	0
CC	1974	1852	122	0	0	0
CD	239	4	177	27	1	30
DEC	2252	2252	0	0	0	0
VV	12170	2424	9516	90	137	3
M	3226	2692	494	25	15	0
LB	23	23	0	0	0	0
LC	1365	1095	270	0	0	0
DEG	2227	2227	0	0	0	0
MSP	150	142	8	0	0	0
P	3879	3493	382	4	0	0
OD	29	29	0	0	0	0
JJ	3271	788	2289	189	5	0
AS	961	961	0	0	0	0
SB	43	43	0	0	0	0
CS	65	14	51	0	0	0
VA	1155	345	754	6	50	0
DT	1372	1219	153	0	0	0
VC	731	731	0	0	0	0
VE	391	379	12	0	0	0
DER	4	4	0	0	0	0
ETC	343	341	2	0	0	0
NN	29890	2099	24226	3351	204	10
FW	8	4	3	1	0	0
PN	864	567	297	0	0	0
DEV	92	92	0	0	0	0

E6 Organization part-of-speech results from People's Daily

337	nz_n_
423	n_n_
158	ns_n_n_n_
130	ns_ns_n_
216	ns_j_
117	n_n_n_
81	nt_j_
92	ns_n_vn_n_
1364	ns_n_
57	n_n_n_n_
160	nt_n_
89	ns_nt_
60	ns_ns_n_n_
130	j_n_
64	ns_vn_n_
70	ns_nz_n_n_
2487	nt_
135	ns_nz_n_
576	ns_n_n_
342	n_j_
69	nz_n_n_

E7 Phrase Tagged People's Daily Data (using Manually generated rules)

```
<DOC>
<HEADER>
<DATE>19980101-01-001</DATE>
```


</TXT>
</BODY>
</DOC>

E8 Correctly tagged People's Daily data

```

<DOC>
<HEADER>
<DATE>19980101-01-001</DATE>
Text originally from People's Daily 1998
Segmentation and POS tagging done by PKU ICL
Modification and LEX tagging done by Daniel Hu at MITRE NLP
April 2001
TAGSET USED: PKU
</HEADER>
<BODY>
<TXT>
<P>
<S>                                </S></P>
<P>
<S><ENAMEX TYPE="ORGANIZATION">    </ENAMEX>                </S></P>
<P>
<S>                                </S></P>
<P>
<S>    <ENAMEX
TYPE="ORGANIZATION">    </ENAMEX>                                </S><S> <EN
AMEX TYPE="ORGANIZATION">    </ENAMEX>                </S></P>
<P>
<S>                                </S></P>
<P>
<S>                                <ENAMEX TYPE="ORGANIZATION">    </ENAMEX> <ENAM
EX TYPE="ORGANIZATION">    </ENAMEX> <ENAMEX
TYPE="ORGANIZATION">    </ENAMEX>
TYPE="ORGANIZATION">    </ENAMEX>
</S></P>
<P>
<S>                                </S><S>
</S><S><ENAMEX
TYPE="ORGANIZATION">    </ENAMEX>                " " " "
</S><S><ENAMEX
TYPE="ORGANIZATION">    </ENAMEX>
</S></P>
<P>
<S>                                </S><S>                " " </S><S>
</S><S>                </S><S>                </S><S>
</S><S>                </S><S>                </S></P>
<P>
<S>                                </S><S>                <ENAMEX
</S><S>                </S><S>                </S><S>
TYPE="ORGANIZATION">    </ENAMEX>                </S></P>
<P>
<S>                                </S><S>                </S><S>
</S><S>                </S><S>                </S></P>
<P>
<S>                                </S><S>                " "
</S></P>
<P>
<S>                                </S><S>                </S><S>                " " " " " "
</S><S>                </S><S>                </S></P>
<P>
<S>                                </S><S>                </S><S>                </S><S>
</S><S>                </S><S>                </S><S>
</S></P>
<P>
<S>                                </S><S>                </S><S>
</S><S>                </S><S>                </S></P>
<P>
<S>                                </S><S>                </S><S>
</S><S>                </S><S>                </S></P>

```

```
<S><ENAMEX  
TYPE="ORGANIZATION"> </ENAMEX>  
  </S><S> <ENAMEX  
TYPE="ORGANIZATION"> </ENAMEX>  
  </S><S> </S><S> </  
S></P>  
<P>  
<S> </S></P>  
<P>  
<S> </S><S> <ENAMEX TYPE="ORGANIZATION"> </ENAMEX> </S></P>  
</TXT>  
</BODY>  
</DOC>
```