DESCRIPTION OF THE KENT RIDGE DIGITAL LABS SYSTEM USED FOR MUC-7

Shihong Yu, Shuanhu Bai and Paul Wu
Kent Ridge Digital Labs
21 Heng Mui Keng Terrace
Singapore 119613
Email: shyu@krdl.org.sg, bai@krdl.org.sg, paulwu@krdl.org.sg

BASIC OF THE SYSTEM

We aim to build a single simple framework for tasks in text information extraction, for which, to a certain extent, the required information can be resolved locally.

Our system is statistics-based. As usual, language model is built from training corpus. This is the so-called learning process. Much effort has been spent to absorb domain knowledge in the language model in a systematic and generic way, because the system is designed not for one particular task, but for general local information extraction.

For the information extraction part (tagging), the system consists of the following modules:

- Sentence segmentor and tokenizer. This module accepts a stream of characters as input, and transforms it into a sequence of sentences and tokens. The way of tokenization can vary with different tasks and domains. For example, most English text is tokenized in the same way, while tokenization in Chinese itself is a research topic.
- Text analyzer. This module provides analysis necessary for the particular task, be it semantic, syntactic, orthographic, etc. This same analyzer is also applied in the learning process.
- Hypothesis generator. The possibilities for each word (token) are determined. Rules can be captured by letting one word have one choice, as is the case in the recognition of time, date, money and percentage terms for the Chinese Named Entity (NE) task. These are identified by pattern matching rules.
- Disambiguation module. This is essentially implementation of Viterbi algorithm.

All the above modules will be described in detail in the following sections.

TEXT INFORMATION EXTRACTION TO TAGGING

First of all, a brief of the modeling of the problem is in order. Each word in text is assigned a tag, information can then be obtained from tags of all words. For example, for the English NE task,

Example 1:
The/- British/- balloon/- ,/- called/- the/- Virgin/- Global/- Challenger/- ,/- is/- to/-

be/- flown/- by/- Richard/PERSON Branson/PERSON ,/- chairman/- of/- Virgin/ORG Atlantic/ORG Airways/ORG ;/-

Grouping all adjacent words with tag PERSON gives a person name, grouping those with tag ORG gives an organization name, etc.

The problem becomes, for any given sequence of words $\boldsymbol{w} = w_1w_2 \dots w_n$, finding the tags $\boldsymbol{t} = t_1t_2 \dots t_n$ correspondingly.

Note that there are different ways of assigning tags. For the above example, tags can also be:

Example 1:
The/- British/- balloon/- ,/- called/- the/- Virgin/- Global/- Challenger/- ,/- is/- to/- be/- flown/- by/- Richard/PERSON-start Branson/PERSON-end ,/- chairman/- of/- Virgin/ ORG-start Atlantic/ORG-continue Airways/ORG-end ;/-

This way, extra information such as common surnames, first names, organization endings (Corp., Inc. etc) and so on can be obtained. It is observed that different tags for a same task make difference. We feel that choosing an appropriate tag set is a problem worthy of careful investigation. Intuitively, a tag set for a particular task must be: sufficient, meaning that the information extracted must be sufficient for the task; and efficient, meaning that there should be no redundant and nonrelevant information.

LEARNING PROCESS: INFORMATION DISTILLATION OF TRAINING CORPUS

Learning Process in General

Careful consideration has been given to study how to absorb domain knowledge in language model(s) in a generic and systematic way. The basic idea is, as much as possible relevant and significant information (to the task) contained in the original corpus should retain in back-off corpora where back-off features are stored, so that correct decisions can be made from the statistics generated from the back-off corpora when they can not be done from the statistics from the original training corpus.

The original training corpus is in the form of word/tag, statistics about words and tags including local contextual information can be obtained. Each word in the corpus is given a back-off feature by the principle that the back-off features of all words should extract the most information from the corpus relevant to the particular task. The information loss is compensated by gain of generosity. A back-off corpus in the form of back-off feature/tag is then generated, and statistics can be obtained in the same manner. The original corpus is processed this way for a certain number of times. Every time, a less descriptive back-off corpus which gains more in generosity is generated, and thus the corresponding statistics.

For example, semantic classes can be used as back-off features for all the words in *Example 1*, which gives the back-off corpus of the following form:

seman1/- seman2/- ... semanM-1/PERSON semanM/PERSON ... semanN-3 /ORG semanN-2/ORG semanN-1/ORG semanN/-

or part-of-speech as back-off features, which gives

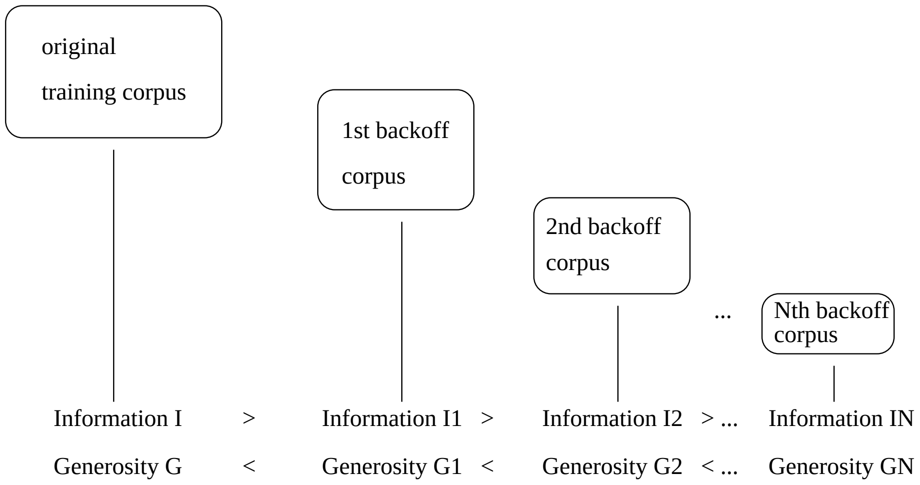

Figure 1: Information Distillation of Training Corpus

pos1/- pos2/- ... posM-1/PERSON - posM/PERSON ... posN-3 /ORG posN-2/ORG posN-1/ORG posn-1/-

The generation of back-off corpora is described by *Figure 1.* The total number of back-off corpora therein is a controllable parameter.

Learning Process for Chinese NE

- Training Corpus and Supporting Resources

  We have a text corpus of about 500,000 words from People Daily and Xinhua News Agency, all of which were manually checked for both word segmentation and part of speech tagging.

  In addition, we have a lexicon of 89,777 words, in which 5351 words are labeled as geographic names, 304 words are people's name and 183 are organization names. 1167 words consist of more than 4 characters. The longest word (meaning "Great Britain and North Ireland United Kingdom") contains 13 characters.

  About 50,000 different words appeared in the 500,000 words corpus.

  We also have three entity name lists: people name list (67,616 entries), location name list (6,451 entries) and organization name list (6190 entries).

- Observation: Problems and Solutions

  1. Intuitively, case information of proper names in English writing system provides good indication about locations and boundaries of entity names. There are successful systems [2] which are built upon this intuition. Unfortunately, the uniformity of character string in Chinese writing system does not contain such information.

     One should look for such analogous indicative characteristics which may be unique in Chinese language.

2. Word in Chinese is a vague concept and there is no clear definition for it. There are boundary ambiguities between words in texts for even human being understanding, and inevitably machine processing. Tokenization, or word segmentation is still a problem in Chinese NLP. Word boundary ambiguities exist not only between commonly used words which are not in entity names, but also between commonly used words and entity names.
3. Besides the uniformity appearance of characters, proper names in Chinese can consist of commonly used words. As a matter of fact, almost all Chinese characters can be a commonly used words themselves, including those in entity names such as people's names, location names, etc.

   Therefore, unlike English, the problem of Chinese entity recognition should not be isolated from the problem of tokenization, or word segmentation.

- Building Language Models

  One level of back-off features, which are also called word classes, are obtained by the following way:

  We extend the idea in the new word detection engine of the integrated model of Chinese word segmentor and part of speech tagger [1]. The idea is to extend the scope of an interested word class of new word, the proper names, into named entities by looking into broader range of constituents. Under this framework, we believe contextual statistics plays important rules in deciding word boundary and predicting the categories of named entities, while local statistics, or information resides within words or entities, can provide evidence for suggesting the appearance of named entity and deciding the validity of these entities. We need to make full use of both contextual and local statistics to recognize these named entities, thus contextual language model and entity models are created.

  The basic process to build the model is like this:

  1. Change the tag set of the part-of-speech tagger by splitting the tag NOUN into more detailed tags related to the particular task, which include the symbolic notions of person, location, organization, date, time, money and percentage.
  2. Replace the tag NOUN in the training corpus with the above extended new tags. Only ambiguous words are manually checked.
  3. Build contextual language model with the training corpus with the new tag set.
  4. Build entity models from the entity name lists. Each entity has its own model.

Learning Process for English NE

- Training Corpus and Supporting Resources

  SGML marked up (for NE task only) Brown corpus and corpus from Wall Street Journal. In total the size of words is 7.2MB, words with SGML-markup is 9.5MB. Supporting resources include the location list, country list, corporation reference list and the people's surname list provided by MUC. Only the single-word entries in these lists are in actual use.

- Observation: Problems and Solutions

  Case information, or more generally, orthographic information, gives good evidence of names, as was observed in [2]. Although things get muddled up when one really gets deep into it: e.g. first words of sentences, words which do not have all normal (lower) case form (e.g. "I"), or words whose cases are changed due to other reasons such as formatting (e.g. titles), being artifacts, etc. Nevertheless, this is an very important information for identifying entity names.

  Prepositions are also helpful, so are common suffixes and prefixes of the entities, such as Corp., Mr., and so on. In general, all such useful information should be somehow sorted out. Word classes tailored for this particular purpose will be ideal.

- Building Language Models

  There are two levels of back-off features represented by word classes.

  For the following words, the two back-off features are the same:

  - Hand-crafted special words for NE task. Each possesses a different word class (represented by word itself). These special words include "I", "the", "past", "pound", "following", "of", "in", "May", etc. In total there are about 100 such words;
  - Words from the supporting resources (as stated in the beginning of this section). Words from a same list possess a same word class.
  - Hand-crafted lists of words, which include week words (Monday, Tuesday, ...), month words (January, February, ...), cardinal numbers (one, two, 1 $\sim$ 31, ...), ordinal numbers (1st, first, 2nd, second, ...), etc.

  For the rest of words, the first level features are word classes provided by a machine auto classification of words, while the second level of features include:

word class	example
oneDigitNum	1
containsDigitAndColon	2:34
containsAlphaDigit	A4
allCaps	KRDL
capPeriod	M.
firstCommonWordInitCap	
firstNonCommonWordIC	
CommonWordInitCap	Department
initCapNotCommonWord	David
mixedCasesWord	ValueJet
charApos	O'clock
allLowerCase	can
compoundWord	ad-hoc

In total, the number of orthographic features is about 30.

To give a sense what information is extracted from the original training corpus, for example, the two back-off sentences for *Example 1* are:

Level 1:
the/- COUN_ADJ/- WordClass1/- ,/- WordClass2/- the/- WordClass3/- WordClass4/ - WordClass5/- ,/- WordClass6/- to/- WordClass7/- WordClass8/- by/- WordClass9/ PERSON WordClass10/PERSON ,/- WordClass11/- of/- WordClass12/ORG Loc/ ORG WordClass13/slash ORG ;/-

Level 2:
the/- COUN_ADJ/- LowerCaseWord/- ,/- LowerCaseWord/- the/- CommonWordInitCap/- CommonWordInitCap/- CommonWordInitCap/- ,/- LowerCaseWord/- to/- LowerCaseWord/- LowerCaseWord/- by/- initCapNotCommonWord/PERSON initCapNotCommonWord/PERSON ,/- LowerCaseWord/- of/- CommonWordInitCap/ORG Loc/ORG CommonWordInitCap/ORG ;/-

Statistics such as the possibilities of CommonWordInitCap (which are NOT first words of sentences) and the corresponding frequencies can be obtained from the second back-off corpus. From our corpus, these are:

Organization	7525
None of the named entities	8493
Location	896
Person	195
Date	8
Money	2

From the above statistics, it's interesting to notice that non-first common words which are initial capitalized have a far more chance to be organization than person (frequencies 7525 vs 195) and location (frequencies 7525 vs 896). This agrees with general observations. Also interesting is that such words have a higher chance not to be any of the seven entities. This comes as a bit surprise. For NLP researchers, though, it may not be a surprise at all. This example also gives a sense how general observations are represented in a precise way.

Further research is to be carried out to justify quantitively the merits of this learning process. Its full potential has yet to be exploited. So far, our experimentation has proved that:

1. Various kinds of text analysis (syntactic, semantic, orthographic, etc) can be incorporated into the same framework in a precise way, which will be used in the information extraction (tagging) stage in the same way;

2. It provides an easy way to absorb human knowledge as well as domain knowledge, and thus customization can be done easily;

3. It gives great flexibility as how to optimize the system.

1 and 2 are somehow clear from the above discussion. Details on the disambiguation module will reveal 3.

DETAILS OF THE SYSTEM MODULES

1. Sentence segmentor and tokenizer: initial tokenization by looking up dictionary for Chinese, standard way for English.
2. Text analyzer. What has been done for training corpus in the learning stage is done here. After the analysis, each word possesses a given number of back-off features.
3. Hypothesis generator.
   - Chinese: based on entities' prefixes, suffixes, trigger words and local context information, guesses are made about possible boundaries of entities and categories of entities. Time, date, money, and percentage are extracted by pattern-matching rules.
   - English: for each word basically look for all the possibilities from the database first. If the word is not found, look for the possibilities of its back-off features.
4. Disambiguation module. Recall that information extraction from word sequence $\boldsymbol{w}$ becomes finding the corresponding tag sequence $\boldsymbol{t}$. In the paradigm of maximum likelihood estimation, the best set of tags $\boldsymbol{t}$ is the one such that $prob(\boldsymbol{t}|\boldsymbol{w}) = \max_{\boldsymbol{t}'} prob(\boldsymbol{t}'|\boldsymbol{w})$. This is equivalently to find $\boldsymbol{t}$ such that $prob(\boldsymbol{tw}) = \max_{\boldsymbol{t}'} prob(\boldsymbol{t}`\boldsymbol{w})$ because $prob(\boldsymbol{t}'|\boldsymbol{w}) = prob(\boldsymbol{tw}')/prob(\boldsymbol{w})$ and $prob(\boldsymbol{w})$ is a constant for any given $\boldsymbol{w}$. The following equality is well-known:

   $$\begin{aligned} prob(\boldsymbol{tw}) &= prob(t_1)\, prob(w_1|t_1)\, prob(t_2|t_1w_1)\, prob(w_2|t_1w_1t_2) \\ &\quad \cdots prob(t_n|t_1w_1\ldots t_{n-1}w_{n-1})\, prob(w_n|t_1w_1\ldots t_{n-1}w_{n-1}t_n). \end{aligned} \tag{1}$$

   Computationally, it is only feasible when some (actually most) dependencies are dropped, for example,

   $$prob(t_k|t_1w_1\ldots t_{k-1}w_{k-1}) \approx prob(t_k|t_{k-1}t_{k-2}), \tag{2}$$
   $$prob(w_k|t_1w_1\ldots t_{k-1}w_{k-1}t_k) \approx prob(w_k|t_kt_{k-1}). \tag{3}$$

   (2) and (3) can be justified by Hidden Markov Modeling for the generation of word sequences.

   As always, Viterbi algorithm is employed to compute the probability (1), given any approximations like (2) and (3). When sparse data problem is encountered, back-off and smoothing strategy can be adopted, e.g.

   $$prob(w_k|t_kt_{k-1}) \quad backoff\ to \rightarrow \quad prob(w_k|t_k), \tag{4}$$

or for unknown words, substitute word in (4) with its back-off features, e.g.

$$
\begin{aligned}
&prob(w_k|t_k t_{k-1}) \quad backoff\ to \rightarrow \quad prob(bof1_k|t_k t_{k-1}) \\
&\quad backoff\ to \rightarrow \quad prob(bof2_k|t_k t_{k-1}) \ \ldots \\
&\quad backoff\ to \rightarrow \quad prob(bofN_k|t_k t_{k-1}) \\
&\quad backoff\ to \rightarrow \quad prob(bof1_k|t_k) \ \ldots \quad backoff\ to \rightarrow \quad prob(bofN_k|t_k),
\end{aligned}
$$

where N is the total number of back-off features for the word.

Note that no smoothing is employed in the above scheme. From this scheme one can see that there exist various ways of back-off and smoothing. This characteristics, as well as the free choices of back-off features, is where the flexibility of the system lies.

Remark. In the actual system, back-off and smoothing schemes are different from the above. The actual schemes are not included because they are more complicated, and yet no systematic experimentation has been done to show that they are better than other options.

PERFORMANCE ANALYSIS

The system currently processes one sentence at a time, and no memory is kept once the sentence is done. Furthermore, due to limitation of time, the guidelines for both Chinese and English NE are not entirely followed, as we didn't have time to read the guidelines carefully!

The F-measures of formal run for Chinese and English are 86.38% and 77.74%, respectively. Given the limited time (less than six months) and resources (three persons, all half time), we are satisfactory with the performance.

* * * CHINESE NE SUMMARY SCORES * * *

	P&R	2P&R	P&2R
F-MEASURES	86.38	84.39	88.46

* * * ENGLISH SUMMARY SCORES * * *

	P&R	2P&R	P&2R
F-MEASURES	77.74	79.06	76.46

FUTURE RESEARCH DIRECTION

Our brief experimentation in Chinese and English Named Entity recognition shows that the system has great potential that deserves further investigation.

1. Modeling of the problem: currently information and knowledge is represented in the form of word/tag. This may pose too much restriction. A better way of representing information and knowledge, in other words, a better modeling of the problem, should be studied.

2. Quantitive justification of the learning process (knowledge distillation) should also be studied. The system should be able to compare different set of back-off features and thus the best one can be chosen.

3. The system provides great flexibility as how to optimize it. The optimization should be done systematicly, rather than trial by trial as is the case for the time being.

References

[1] S. Bai, *An Integrated Model of Chinese Word Segmentation and Part of Speech Tagging,* Advances and Applications on Computational Linguistics (1995), Tsinghua University Press.

[2] D.M. Bikel, S. Miller, R. Schwartz and R. Weischedel, *Nymble: a High-Performance Learning Name-finder.*