Information extraction using finite state automata and syllable n-grams in a mobile environment

Choong-Nyoung Seon
Computer Science and Engineering
Sogang University
Seoul, Korea
wilowisp@gmail.com

Harksoo Kim
Computer and Communications Engineering
Kangwon National University
Chuncheon, Korea
nlpdrkim@kangwon.ac.kr

Jungyun Seo
Computer Science and Engineering
Sogang University
Seoul, Korea
seojy@sogang.ac.kr

Abstract

We propose an information extraction system that is designed for mobile devices with low hardware resources. The proposed system extracts temporal instances (dates and times) and named instances (locations and topics) from Korean short messages in an appointment management domain. To efficiently extract temporal instances with limited numbers of surface forms, the proposed system uses well-refined finite state automata. To effectively extract various surface forms of named instances with low hardware resources, the proposed system uses a modified HMM based on syllable *n*-grams. In the experiment on instance boundary labeling, the proposed system showed better performances than traditional classifiers.

1 Introduction

Recently, many people access various multi-media contents using mobile devices such as a cellular phone and a PDA (personal digital assistant). Accordingly, users' requests on NLP (natural language processing) are increasing because they want to easily and simply look up the multi-media contents. Information extraction is one of useful applications in NLP that helps users to easily access core information in a large amount of free texts. Unfortunately, it is not easy to implement an information extraction system in mobile devices because target texts include many morphological variations (*e.g.* blank omission, typos, word abbreviation) and mobile devices have many hardware limitations (*e.g.* a small volume of a main memory and the absence of an arithmetic logic unit for floating-point calculation)

There are some researches on information extraction from short messages in a mobile device, and Cooper's research (Cooper, 2005) is representative. Cooper predefined various syntactic patterns with placeholders and matched an input message against the syntactic patterns. Then, he extracted texts in the placeholders and assigned them the attribute name of the placeholders. This method has some advantages like easy implementation and fast response time. However, it is inadequate to apply Cooper's method to languages with partially-free word-order like Korean and Japanese because a huge amount of syntactic patterns should be predefined according as the degree of freedom on word order increases. Kang (2004) proposed a NLIDB (natural language interface to database) system using lightweight shallow NLP techniques. Kang raised problems of deep NLP techniques such as low portability and error-proneness. Kang proposed a lightweight approach to natural language interfaces, where translation knowledge is semi-automatically acquired and user questions are only syntactically analyzed. Although Kang's method showed good performances in spite of using shallow NLP techniques, it is difficult to apply his method to mobile devices because his method still needs a morphological analyzer with a large size of dictionary. In this paper, we propose an information extraction system that is designed for mobile devices with low hardware resources. The proposed system extracts appointment-related information (*i.e.* dates, times, locations, and topics) from Korean short messages.

This paper is organized as follows. In section 2, we proposed an information extraction system for a mobile device in an appointment domain. In sec-

Proceedings of the ACL-08: HLT Workshop on Mobile Language Processing, pages 13–18,
Columbus, Ohio, USA, June 2008.

tion 3, we explain experimental setup and report some experimental results. Finally, we draw some conclusions in section 4.

2 Lightweight information extraction system

Figure 1 shows an overall architecture of the proposed system.

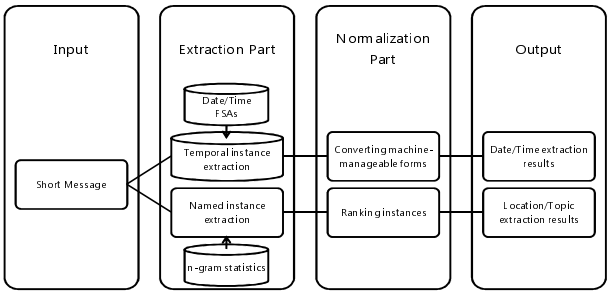

Figure 1. The system architecture

As shown in Figure 1, the proposed system consists of an extraction part and a normalization part. In the extraction part, the proposed system first extracts temporal instance candidates (*i.e.* dates and times) using FSA (finite-state automata). Then, the proposed system extracts named instance candidates (*i.e.* locations and topics) using syllable *n*-grams. Finally, the proposed system ranks the extracted instances and selects the highest one per target category. In the normalization part, the proposed system converts the temporal instances into suitable forms.

2.1 Information extraction using finite state automata

Although short messages in an appointment domain often include many incorrect words, temporal instances like dates and times are expressed as correct as possible because they are very important to appointment management. In addition, temporal instances are expressed in tractable numbers of surface forms in order to make message receivers easily be understood. In MUC-7, these kinds of temporal instances are called TIMEX (Chinchor, 1998), and it has known that TIMEX can be easily extracted by using FSA (Srihari, 2001). Based on these previous works, the proposed system extracts temporal instances from short messages by using FSA, as shown in Figure 2.

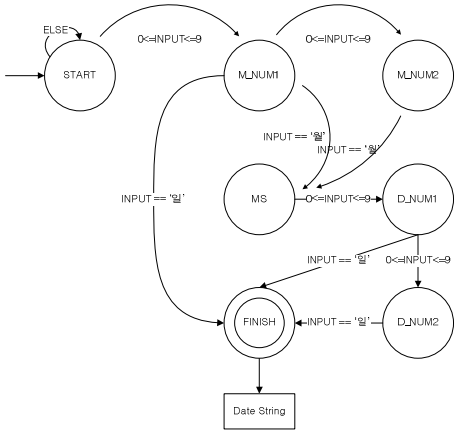

Figure 2. An example of FSA for date extraction

2.2 Information extraction using statistical syllable n-grams

Unlike dates and times, locations and topics not only have various surface forms, but also their constituent words are not included in a closed set. In MUC-7, these kinds of named entities are called NAMEX (Chinchor, 1998), and many researches on NAMEX have been performed by using rules and statistics. Generally, rule-based methods show high precisions but they have a weak point that it is hard to maintain a system when new words are continuously added to the system (Goh, 2003). Statistical methods guarantee reasonable performances but they need large-scale language resource and complex floating point operations. Therefore, it is not suitable to apply previous traditional approaches to mobile devices with many hardware limitations. To resolve this problem, we propose a statistical model based on syllable *n*-grams, as shown in Figure 3.

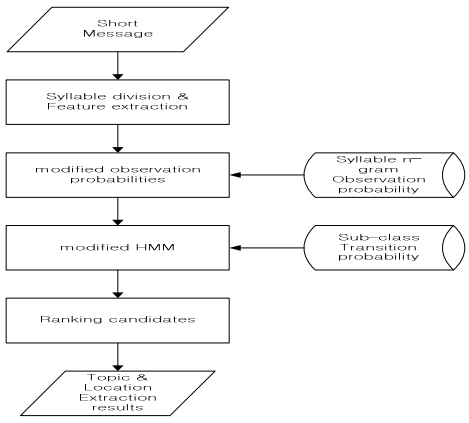

Figure 3. Statistical information extraction

The extraction of named instances has two kinds of problems; a instance boundary detection problem and a category assigning problem. If we can use a conventional morphological analyzer, the instance boundary detection problem is not big. However, it is not easy to use a morphological analyzer in a mobile device because of hardware limitations and users' writing habitations. Users often ignore word spacing and this habitation lowers the performance of the morphological analyzer. To resolve this problem, we adopt a syllable *n*-gram model that performs well in word boundary detection for languages like Chinese with no spacing between words (Goh, 2003; Ha, 2004). We first define 9 labels that represent boundaries of named instance candidates by adopting BIO (begin, inner, and outer) annotation scheme, as shown in Table 1 (Hong, 2005; Uchimoto, 2000).

Tag	Description	Tag	Description
LB	Begin of a location	TB	Begin of a topic
LI	Inner of a location	TI	Inner of a topic
LE	End of a location	TE	End of a topic
LS	A single-syllable location	TS	A single-syllable topic
OT	Other syllable		

Table 1. The definition of instance boundary labels

Then, based on a modified HMM (hidden Markov model), the proposed system assigns boundary labels to each syllable in an input message, as follows.

Let $S_{1,n}$ denote a message which consists of a sequence of *n* syllable, $s_1, s_2, ..., s_n$, and let $L_{1,n}$ denote the boundary label sequence, $l_1, l_2, ..., l_n$, of $S_{1,n}$. Then, the label annotation problem can be formally defined as finding $L_{1,n}$ which is the result of Equation (1).

$$\begin{aligned} L(S_{1,n}) &\overset{def}{=} \arg\max_{L_{1,n}} P(L_{1,n} \mid S_{1,n}) \\ &= \arg\max_{L_{1,n}} \frac{P(L_{1,n}, S_{1,n})}{P(S_{1,n})} \\ &= \arg\max_{L_{1,n}} P(L_{1,n}, S_{1,n}) \end{aligned} \quad (1)$$

In Equation (1), we dropped $P(S_{1,n})$ as it is constant for all $L_{1,n}$. Next, we break Equation (1) into bite-size pieces about which we can collect statistics, as shown in Equation (2).

$$P(L_{1,n}, S_{1,n}) = \prod_{i=1}^{n} P(s_i \mid l_{1,i}, s_{1,i-1}) P(l_i \mid l_{1,i-1}, s_{1,i-1}) \quad (2)$$

We simplify Equation (2) by making the following two assumptions: one is that the current boundary label is only dependent on the previous boundary label, and the other is that current boundary label is affected by its contextual features.

$$P(L_{1,n}, S_{1,n}) = \prod_{i=1}^{n} P^*(s_i \mid l_i) P(l_i \mid l_{i-1}) \quad (3)$$

In Equation (3), $P^*(s_i \mid l_i)$ is a modified observation probability that is adopted from a class probability in naïve Bayesian classification (Zheng, 1998) as shown in Equation (4). The reason why we modify an original observation probability $P(s_i \mid l_i)$ in HMM is its sparseness that is caused by a size limitation of training corpus in a mobile environment.

$$P^*(s_i \mid l_i) = \frac{1}{Z} P(l_i) \prod_{j=1}^{f} P(s_{ij} \mid l_i) \quad (4)$$

In Equation (4), f is the number of contextual features, and s_{ij} is the *j*th feature of the *i*th syllable. Z is a normalizing factor. Table 2 shows the contextual features that the proposed system uses.

Feature	Composition	Meaning
s_{i1}	s_i	The current syllable
s_{i2}	$s_{i-1}s_i$	The previous syllable and the current syllable
s_{i3}	$s_i s_{i+1}$	The current syllable and the next syllable

Table 2. The composition of contextual features

In Equation (1), the max scores are calculated by using the well-known Viterbi algorithm (Forney, 1973).

After performing instance boundary labeling, the proposed system extracts syllable sequences labeled with the same named categories. For example, if a syllable sequence is labeled with 'TS OT LB LI LI', the proposed system extracts the sub-sequence of syllables labeled with 'LB LI LI',

as a location candidate. Then, the proposed system ranks the extracted instance candidates by using some information such as position, length, and a degree of completion, as shown in Equation (5).

$$Rank(NI_i) = \alpha \cdot Position_i + \beta \cdot Length_i + \gamma \cdot Completion_i \quad (5)$$

In Equation (5), $Position_i$ means the distance from the beginning of input message to the *i*th named instance candidate NI_i. In Korean, important words tend to appear in the latter part of a message. Therefore, we assume that the latter part an instance candidate appears in, the more important the instance candidate is. $Length_i$ means the length of an instance candidate. We assume that the longer an instance candidate include is, the more informative the instance candidate is. $Completion_i$ means whether a sequence of instance boundary labels is complete. We assume that instance candidates with complete label sequences are more informative. To check the degree of completion, the proposed system uses FSA, as shown in Figure 4. In the training corpus, every transition is legal. Therefore most of candidates were satisfied the completion condition. However, sometimes the completion condition is not satisfied, when the candidate was extracted from the boundary of a sentence. Accordingly the condition gave an effect to the rank.

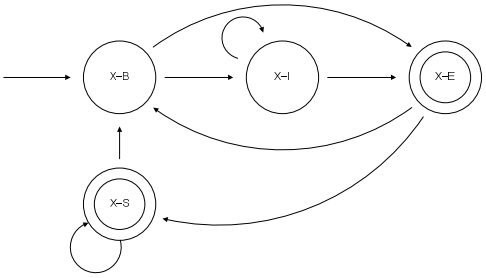

Figure 4. The FSA for checking label completion

In the experiments, we set α, β, and γ to 1, 2, and 10, respectively.

2.3 Normalization of temporal instances

It is inadequate for the proposed system to use the extracted temporal instances as database instances without any processing because the temporal instances consist of various forms of human-readable strings like 'January 24, 2008'. Therefore, the proposed system should normalize the temporal instances into machine-manageable forms like '20080124'. However, the normalization is not easy because temporal instances often include the relative information like 'this Sunday' and 'after two days'. To resolve this problem, the proposed system converts relative temporal instances into absolute temporal instances by using a message arrival time. For example, if a message includes the temporal instance 'after two days', the proposed system checks arrival time information of the message. Then, the proposed system adds a date in the arrival time information to two days. Figure 5 shows an example of date normalization.

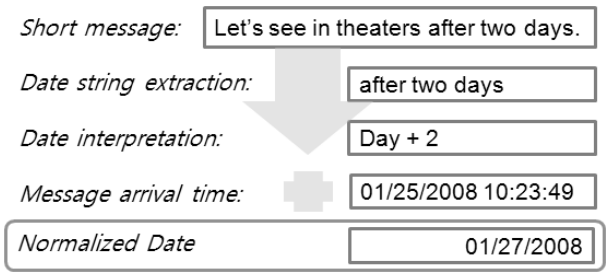

Figure 5. An example of date normalization

3 Evaluation

3.1 Data sets and experimental settings

We collected 6,190 short messages simulated in an appointment scheduling domain. These messages contain 4,686 locations and 4,836 topics. Each message is manually annotated with the boundary labels in Table 1. The manual annotation was done by 2 graduate students majoring in natural language processing and post-processed by a student in a doctoral course for consistency. In order to experiment the proposed system, we divided the annotated messages into the training corpus and the testing corpus by a ratio of four (4,952 messages) to one (1,238 messages). Then, we performed 5-fold cross validation and used a precision, a recall rate, and a F1-measure as performance measures. In this paper, we did not evaluate the performances on the temporal instance extraction because performances of the proposed method are fully dependent on the coverage of pre-constructed FSA.

3.2 Experimental results

To choose the proper size of language models in a mobile environment, we evaluated performance variations of the proposed system, as shown in Figure 6.

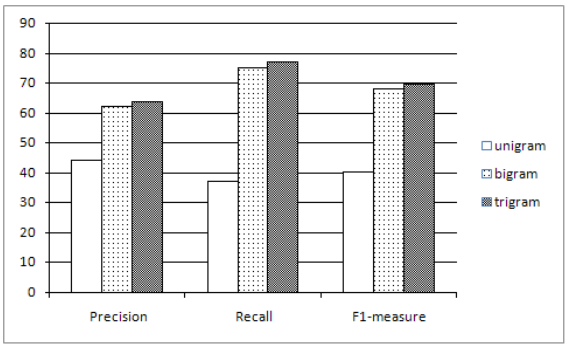

Figure 6. The performance variations according to the size of language models

As shown in Figure 6, the system using syllable unigrams showed much lower performances than the systems using syllable bigrams or syllable trigrams.

	Bigram	Trigram
# of features	54,711	158,525
Size of DB	1.33M	2.83M

Table 3. Space requirements of language models.

However, as shown in the Table 3, although the number of syllable trigrams was three times larger than the number of syllable bigrams, the difference of performances between the system using syllable bigrams and the system using syllable trigrams was not big (about 1%). Based on this experimental result, we conclude that the combination of syllable unigrams and syllable bigrams, as shown in Table 2, is the most suitable language model for mobile devices with low hardware resources.

To evaluate the proposed system, we calculated two types of performances. One is boundary labeling performances that measure whether the proposed system can correctly annotate a test corpus with boundary labels in Table 1. The other is extraction performances that measure whether the proposed system can correctly extract named instances from a test corpus by using Equation (5). Table 4 shows the boundary labeling performances of the proposed system in comparisons with those of representative classifiers.

Model	Precision	Recall rate	F1-measure
NB	62.74%	75.17%	68.34%
SVM	67.29%	67.58%	67.37%
CRF	70.98%	66.27%	68.45%
Proposed system	74.81%	77.20%	75.91%

Table 4. The comparison of boundary labeling performances

In Table 4, NB is a classifier using naïve Bayesian statistics, and SVM is a classifier using a support vector machine. CRF is a classifier using conditional random fields. As shown in Table 4, the proposed system outperformed the comparative models in all measures. Based on this fact, we think that the modified HMM may be more effective in a labeling sequence problem.

Table 5 shows the extraction performances of the proposed system. In Table 5, the reason why the performances on the topic extraction are lower is that topic instances can consist of more various syllables (e.g. the topic instance, 'a meeting in Samsung Research Center', includes the location, 'Samsung Research Center').

Category	Precision	Recall rate	F1-measure
Location	79.37%	76.33%	77.78%
Topic	58.54%	55.20%	56.72%

Table 5. The extraction performances

Table 6 shows performance variations according as the parameters in Equation (5) are changed. As shown in Table 6, the differences between performances are not big, and the proposed model showed the best performance at (α=1, β=2, γ=5) or (α=1, β=2, γ=10). On the basis of this experiments, we set α, β, and γ to 1, 2, and 5, respectively.

(α,β,γ)	Precision of Location	Recall rate of Location	F1-measure of Location
(1,1,1)	79.23%	76.20%	77.65%
(1,1,5)	79.28%	76.24%	77.69%

(1,1,10)	79.30%	76.26%	77.71%
(1,2,5)	**79.37%**	**76.33%**	**77.78%**
(1,2,10)	**79.37%**	**76.33%**	**77.78%**
(α,β,γ)	Precision of Topic	Recall rate of Topic	F1-measure of Topic
(1,1,1)	58.09%	54.76%	56.28%
(1,1,5)	58.09%	54.76%	56.28%
(1,1,10)	58.11%	54.78%	56.30%
(1,2,5)	**58.54%**	**55.20%**	**56.72%**
(1,2,10)	**58.54%**	**55.20%**	**56.72%**

Table 6. The performance variations according to parameter changes

To evaluate usefulness of the proposed model in a real mobile phone environment, we measured an average response time of 100 short messages in a mobile phone with XSCALE PXA270 CPU, 51.26MB memory, and Windows mobile 5.0. We obtained an average response time of 0.0532 seconds.

4 Conclusion

We proposed an information extraction system for a mobile device in an appointment management domain. The proposed system efficiently extracts temporal instances with limited numbers of surface forms by using FSA. To effectively extract various surface forms of named instances with low hardware resources, the proposed system uses a modified HMM based on syllable n-grams. In the experiment on instance boundary labeling, the proposed system outperformed traditional classifiers that showed good performances in a labeling sequence problem. On the experimental basis, we think that the proposed method is very suitable for information extraction applications with many hardware limitations.

Acknowledgments

This research (paper) was funded by Samsung Electronics.

5 Reference

Chooi Ling Goh, Masayuki Asahara, Yuji Matsumoto. 2003. Chinese unknown word identification using character-based tagging and chunking. *Proceedings of ACL-2003 Interactive Posters and Demonstrations*, 197-200.

G. David Forney, JR. 1973. The Viterbi Algorithm *Proceedings of the IEEE,* 61(3):268-278.

Hong Shen, Anoop Sarkar. 2005. Voting Between Multiple Data Representations for Text Chunking. *Canadian Conference on AI 2005*. 389-400.

In-Su Kang, Seung-Hoon Na, Jong-Hyeok Lee, Gijoo Yang. 2004. Lightweight Natural Language Database Interfaces. *Proceedings of the 9th International Conference on Application of Natural Language to Information Systems*. 76-88.

Juhong Ha, Yu Zheng, Byeongchang Kim, Gary Geunbae Lee, Yoon-Suk Seong. 2004. High Speed Unknown Word Prediction Using Support Vector Machine for Chinese Text-to-Speech Systems. IJCNLP:509-517

Kiyotaka Uchimoto, Qing Ma, Masaki Murata, Hiromi Ozaku, and Hitoshi Isahara. Named Entity Extraction Based on A Maximum Entropy Model and Transformation Rules. In Proceedings of the 38th Annual Meeting of Association for Computational Linguistics

Nancy A. Chinchor. 1998. MUC-7 named entity task definition, Proceedings of the Seventh Message Understanding Conference.

Richard Cooper, Sajjad Ali, Chenlan Bi, 2005. Extracting Information from Short Messages, NLDB 2005, LNCS 3513, pp. 388-391.

Rohini Srihari, Cheng Niu, Wei Li. 2001. A hybrid approach for named entity and sub-type tagging. In Proc. 6th Applied Natural Language Processing Conference.

Zijian Zheng. Naive Bayesian classifier committees. Proceedings of the 10th European Conference on Machine Learning. Berlin: Springet-Verlag (1998) 196-207.

SVM_light: http://svmlight.joachims.org/

CRF++: http://crfpp.sourceforge.net/