Journal of The Korea Society of Computer and Information
Vol. 22 No. 12, pp. 35-48, December 2017
www.ksci.re.kr
https://doi.org/10.9708/jksci.2017.22.12.035

ExoTime: Temporal Information Extraction from Korean Texts Using Knowledge Base

Young-Seob Jeong*, Chae-Gyun Lim**, Ho-Jin Choi***

Abstract

Extracting temporal information from documents is becoming more important, because it can be used to various applications such as Question-Answering (QA) systems, Recommendation systems, or Information Retrieval (IR) systems. Most previous studies only focus on English documents, and they are not applicable to the other languages due to the inherent characteristics of languages. In this paper, we propose a new system, named ExoTime, designed to extract temporal information from Korean documents. The ExoTime adopts an external Knowledge Base (KB) in order to achieve better prediction performance, and it also applies a bagging method to the temporal relation prediction. We show that the effectiveness of the proposed approaches by empirical results using Korean TimeBank. The ExoTime system works as a part of ExoBrain that is an artificial intelligent QA system.

▸Keyword: temporal information extraction, temporal expression, temporal relation, Korean TimeBank

I. Introduction

Documents usually contain temporal information (e.g., May 2009, yesterday) that is useful for developing various applications such as Question-Answering (QA) systems, Recommendation systems, or Information-Retrieval (IR) systems. In most cases, the temporal information within the documents is not represented in a structured way, which makes it difficult to use. Thus, it is necessary to develop a method that finds such information within documents and converts it to a structured form comprehensible by computers. For example, given the sentence “He arrived at 3 pm”, the temporal expression ‘3 pm’ can be found and could be converted into a structured form (e.g., 15:00). This is the task of temporal information extraction.

Although there have been many studies about the extraction of temporal information, most of them are not applicable to Korean language because their target language is English. Korean language is an agglutinative language, and some elements appear very often because subjects and objects may be dropped to avoid duplication. It also has a complex system of numerical expression. These characteristics of Korean language make it difficult to apply existing studies that target the English language to Korean language.

For the task of temporal information extraction from Korean texts, there are three challenging issues to address. First, the existing annotation languages of the temporal information are not compatible with Korean

• First Author: Young-Seob Jeong, Corresponding Author: Ho-Jin Choi
*Young-Seob Jeong (bytecell@sch.ac.kr), Dept. of Big Data Engineering, Soonchunhyang University
**Chae-Gyun Lim (rayote@kaist.ac.kr), School of Computing, KAIST
***Ho-Jin Choi (hojinc@kaist.ac.kr), School of Computing, KAIST
• Received: 2017. 10. 21, Revised: 2017. 11. 03, Accepted: 2017. 12. 04.
• This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP; Ministry of Science, ICT & Future Planning) (No. 2017017836). This work was supported by the Soonchunhyang University Research Fund.

language. Although ISO-TimeML was designed to incorporate the characteristics of various languages, it does not provide a way to incorporate some important characteristics of Korean language. For example, ISO-TimeML does not consider the calendar type (e.g., Gregorian, Lunar), whereas temporal expressions using the lunar calendar often appear in Korean texts. Second, there is no sufficient dataset. It is obvious that the shortage of datasets may harm the system performance, especially when the system employs the data-driven approach. Third, there are only a few studies about extraction of temporal information from Korean texts. Fortunately, a temporal information extraction system for Korean was recently developed using a newly constructed dataset [1]. Based on this, in this paper, we propose an extension of the system, namely ExoTime. The main difference between the ExoTime system and the previous system is that ExoTime employs an External Knowledge-Base (EKB) to achieve better normalization performance of temporal expressions. It also employs a bagging method for the extraction of temporal relation, in order to improve the extraction performance, especially regarding recall. The ExoTime system works as a part of ExoBrain that is an artificial intelligent QA system.

The rest of this paper is organized as follows. Section 2 provides a preliminary background of related studies. Section 3 details the research objective and the proposed approach in detail. Section 4 describes the experiments and results. Finally, Section 5 presents the conclusions and indicates future work.

II. Related Work

Because the extraction of temporal information became more important, there have been some studies about the annotation languages [2-4]. Based on the annotation languages, TempEval shared tasks have appeared [5-7], and there were many related studies [8-14]. Although these studies achieved fairly good results in terms of the extraction performance, they are applicable only to their target languages. The target language was mostly English, which has quite different characteristics from Korean. Korean is an agglutinative language, where verbs are formed by attaching various endings to a stem. There are usually multiple morphemes for each token, and empty elements appear very often because subjects and objects are dropped to avoid duplication. Korean has a complex system of numerical expression. For instance, the number 30 can be represented as '30', '삼십 [sam-sib]', or '서른 [seo-run]'. These characteristics of Korean language make it more difficult to adapt existing studies with English as the target language to Korean text. Thus, it is necessary to develop a system by which to extract temporal information from Korean texts.

There have been few studies about temporal information extraction from Korean texts. In [15], a method for extracting temporal expressions was described, for which the annotation was based on timex2. It provided a distributional comparison of different types of temporal expressions between English and Korean. The paper also discussed some issues associated with the annotation process with Korean, and reported that rote learning approaches could be useful when language-specific features are available. Seon, Kang, and Seo (2010) defined a set of lexicalized rules for extracting temporal expressions and values. The authors reported that the proposed method achieved about 93% accuracy using manually annotated SMS (Short Message Service) datasets [16]. Kim and Choi (2011) proposed a system for extracting temporal information from clinical narratives [17]. It used finite state automata (FSA) and a predefined dictionary to recognize temporal expressions, and attempted to get normalized temporal values by checking the temporal expressions with regular expression patterns. Angeli and Uszkoreit (2013) proposed a method for extracting temporal expressions [18]. It employed discriminative parsing to encode a language-flexible representation of temporal expressions. The proposed parsing model was trained using a weakly supervised bootstrapping approach without the need for language expertise. Its effectiveness was proven by experiments with TempEval-2 datasets of six languages: Chinese, English, French, Italian, Spanish, and Korean. The performance of extraction of temporal values with Korean was 42% accuracy. This implies that it basically requires language-specific feature engineering to achieve satisfactory performance. In [1], the first system that extracts temporal expressions, event expressions, and temporal relations, was proposed. It used a hybrid variation of the rule-based approach and the data-driven approach, and also proposed a probabilistic model generating complementary features, in order to improve the overall performance.

In this paper, we propose a new system named ExoTime. This is an extension of our previous work [1] and it improved the performance of machine-learning algorithms and hand-crafted rules. It also allows the connection with an external Knowledge-Base (KB), and proves its usefulness by experiments with an imaginary KB, named Oracle, which is expected to have all necessary knowledge.

III. Temporal Information Extraction from Korean Texts

1. Task Description

The purpose of ExoTime is to extract temporal information from Korean texts. We adopt a part of Korean Time Markup Language (Korean TimeML) [19] as an annotation language. The input of ExoTime is just the Korean texts written using natural language, and the output is the annotated tags. To be more specific, the annotated tags include the *timex3*, event, *makeinstance,* and *tlink* tags. The summarization of the *timex3* attributes to extract, is described in Table 1.

Table 1. Summary of *timex3* attributes to be extracted

Attributes	Explanation
id	*timex3* tag ID
type	Type (DATE, TIME, DURATION, SET)
e_begin	Beginning token index of the extent
e_end	Ending token index of the extent
begin	Beginning character index of the extent
end	Ending character index of the extent
text	The string within the extent
value	Normalized temporal value based on ISO-8601 (e.g., 1986-04-04)
beginPoint	*timex3* tag ID having the begin time of the duration (used for DURATION type)
endPoint	*timex3* tag ID having the end time of the duration (used for DURATION type)
freq	Frequency value (used for SET type)
prd	Period value (used for SET type)
quant	Quantity value (used for SET type)
mod	Type of vague representation (e.g., beginning, around March)
calendar	Calendar type (e.g., GREGORIAN, LUNAR, JULIAN, ...)

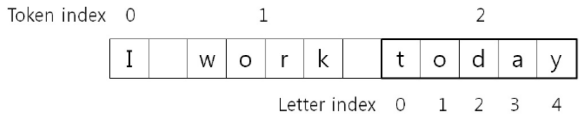

Fig. 1. Example of extent representation by word indices and character indices

There are four attributes used to indicate the tag extent: *e_begin, e_end, begin,* and *end.* The Korean TimeBank is annotated at a character level using the four attributes, where *e_begin,* and *e_end* indicate the word indices; and *begin* and *end* indicate the character indices. For the sentence “I work today” in Fig. 1, there will be a *timex3* tag indicated by *text* ‘today’, where *e_begin* = 2, *e_end* = 2, *begin* = 0, and *end* = 4. Due to the characteristics of Korean, sometimes only a part of a word is extracted as an extent. For the sentence “나는 5월까지 일했다 (I worked until May)”, the word ‘5월’ (in English, ‘May’) should be the extent of *timex3* tag. If we assume that the extent is determined in the word-level (i.e., separated by blank), then the word ‘5월까지’ (in English, ‘until May’) will be the extent of the *timex3* tag. In order to exactly represent the extent ‘5월’ which is a part of the word ‘5월까지’, we introduced the four attributes (*e_begin, e_end, begin,* and *end*). For the *timex3* tag of the extent ‘5월’, the four attributes will be *e_begin* = 1, *e_end* = 1, *begin* = 0, and *end* = 1. This character-level annotation also makes the dataset independent of the morpheme tag-set and morphological analysis.

The attribute *type* can have four values: DATE, TIME, DURATION, and SET. The DATE value means that the tag extent has a date expression (e.g., ‘8 March’), while the TIME value means that the tag extent has more specific information than just a date (e.g., ‘morning’, ‘10:00 am’). The DURATION value means that the tag is a duration or an interval, and the SET value associates temporal patterns (e.g., ‘twice a week’). The attribute *text* is just a string of the tag extent. The attribute value has a normalized temporal value which follows ISO-8601. For the sentence “We were there 3 January, 2016”, there will be a *timex3* tag whose *text* = ‘3 January, 2016’ and *value* = ‘2016-01-03’. The attributes *beginPoint* and *endPoint* are used to indicate the beginning and ending of a temporal interval when *type* is DURATION. The attributes *freq, prd,* and *quant* are used to represent information about temporal patterns when *type* is SET. The attribute *mod* is used to represent vague temporal information (e.g., beginning, about, around). The attribute *calendar* is used to indicate the calendar type, where the default value is GREGORIAN. More details of the attribute definitions can be found in [19].

Table 2. Summary of *event* attributes to be extracted

Attributes	Explanation
id	*event* tag ID
class	Event class (REPORTING, PERCEPTION, I_ACTION, I_STATE, STATE, OCCURRENCE)
e_begin	Beginning token index of the extent
e_end	Ending token index of the extent
begin	Beginning character index of the extent
end	Ending character index of the extent
text	The string within the extent

Table 3. Summary of *makeinstance* attributes to be extracted

Attributes	Explanation
id	*makeinstance* tag ID
eventID	Corresponding *event* tag ID
polarity	Boolean attribute (e.g., POS, NEG) representing the polarity associated with the *event* tag
tense	Tense of the text (PAST, PRESENT, FUTURE, NONE)
POS	Part-Of-Speech (POS) tag (ADJECTIVE, NOUN, VERB, PREPOSITION, OTHER)
modality	Indication of conjectural expression (CONJECTURAL, NONE)
cardinality	The number of event occurrences (default:1)

The summarization of event attributes and *makeinstance* attributes to extract are described in Table 2 and Table 3, respectively. The event tag plays the role of event token, while the *makeinstance* tag plays a role of event instance. That is, the event tag can be thought of as a template, and the *makeinstance* tag is an instance of the template. As with the *timex3* tag, there are four event attributes to indicate the extent: *e_begin, e_end, begin,* and *end.* The attribute *text* is just a string of the tag extent. The attribute *eventID* of the *makeinstance* tag is the *id* of the corresponding *event* tag. The attribute *polarity* indicates whether the occurrence of an event is positive or not. The attribute *tense* represents the tense of the *text* of the corresponding *event* tag, and the attribute *POS* is the Part-Of-Speech (POS) tag of the *text.* The attribute *modality* indicates whether the event expression is conjectural or not, while the attribute *cardinality* is the number of event occurrences.

Table 4. Summary of *tlink* attributes to be extracted

Attributes	Explanation
id	*tlink* tag ID
eventInstanceID	ID of the first argument tag (when it is *makeinstance* tag)
timeID	ID of the first argument tag (when it is *timex3* tag)
relatedToEventInstance	ID of the second argument tag (when it is *makeinstance* tag)
relatedToTime	ID of the second argument tag (when it is *timex3* tag)
relType	Link (relation) type (BEFORE, AFTER, DURING, DURING_INV, SIMULTANEOUS, INCLUDES, IDENTITY, BEGINS, ENDS, OVERLAP)

The *tlink* is a linkage between two other argument tags. TT tlink is a linkage between two *timex3* tags; MM *tlink* is a linkage between two *makeinstance* tags. TM *tlink* is a linkage between a *timex3* tag and a *makeinstance* tag. Note that the *makeinstance* tag is used as an argument of the *tlink* tag, while the *event* tag is just a template of the *makeinstance* tags. The summarization of *tlink* attributes to extract is described in Table 4. The attributes *eventInstanceID* and *timeID* are used to denote the first argument, while the attributes *relatedToEventInstance* and *relatedToTime* are used to denote the second argument. The attribute *relType* represents the type of temporal relation between the two arguments.

2. ExoTime System

The ExoTime system is designed to extract *timex3* tags, *event* tags, *makeinstance* tags, and *tlink* tags. The overall process, as depicted in Fig. 2, is an extension of our previous system [1], where the solid line represents the training process and the dotted line represents the testing process. The main difference between ExoTime system and our previous system is that the External KB (EKB) is introduced as depicted in the upright corner of the figure. The previous system could not fill the *value* of the *timex3* tags that require some external knowledge. For the sentence “He played an active part in the background during Napoleonic times”, the previous system could not fill the *value* of the *timex3* tag whose *text* = ‘Napoleonic times’. The ExoTime system employs the EKB to resolve such limitations. The new system also

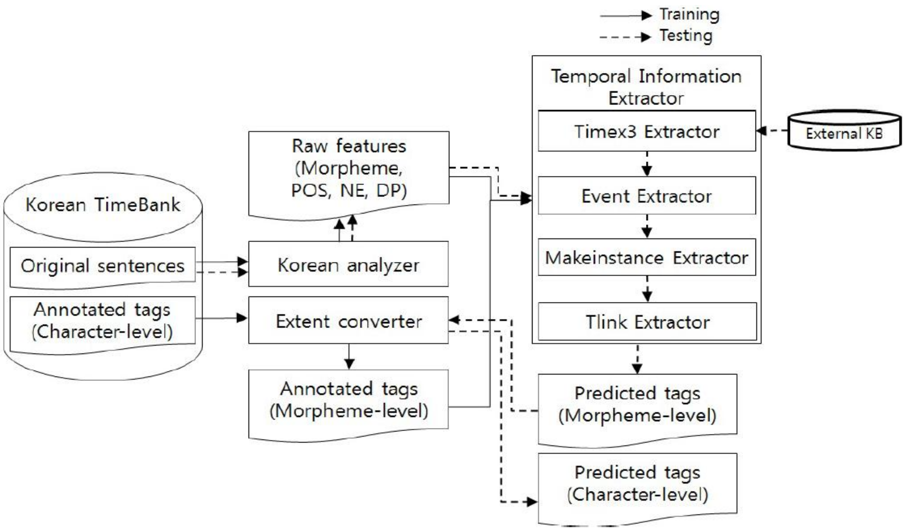

Fig. 2. Overall process of ExoTime system

adopts a bagging strategy to improve the recall of *tlink* extraction.

The ExoTime system employs a Korean analyzer that takes Korean text as input and generates several raw features as output, as depicted in the center of the figure. The raw features are the results of morphological analysis, Part-Of-Speech (POS) tags, Named-Entity (NE) tags, and results of dependency parsing [20]. Based on the generated raw features, a set of rules is defined as the rule-based approach, and a set of features are defined to train the machine-learning models (e.g., Maximum Entropy Model, Support Vector Machine, Conditional Random Fields, and Logistic Regression).

There are four sub-extractors: *timex3* extractor, *event* extractor, *makeinstance* extractor, and *tlink* extractor. The four sub-extractors are designed to work in a cascade. Although the event extractor is performed after the *timex3* extractor, the two extractors work independently. When the event extraction is completed, the *makeinstance* extractor works based on the predicted *event* tags. Because the *tlink* is a linkage between the other tags, the *tlink* extractor works based on the other predicted *event* tags, *makeinstance* tags, and *timex3* tags. Thus, the performance of the tlink extractor will strongly depend on the other extractors. These four sub-extractors, as a whole, give predicted tags as output, where the tags are represented at a morpheme level.

The ExoTime system is designed to work at a morpheme level, while the Korean TimeBank is annotated at a character level because the character-level annotation makes it easier to distribute/share the dataset. As a bridge between the system and the dataset, the extent converter at the center of the figure changes the morpheme-level tags to character-level (and vice versa) by checking the ASCII values of each character and each morpheme. Using the extent converter allows the system to be applicable to any dataset annotated at a character level. During the training process, the annotated tags of Korean TimeBank are converted to morpheme-level through the extent converter, and used to train the system.

3. TIMEX3 Extractor

The goal of the *timex3* extractor is to predict whether each morpheme belongs to the extent of a *timex3* tag or not, and to find appropriate attributes of the tag. More precisely, it predicts the *type* of each morpheme, where there are five *types* of *timex3* tag: DATE, TIME, DURATION, SET, and NONE. The NONE indicates that the corresponding morpheme does not belong to the extent of a *timex3* tag. This is essentially a morpheme-level classification over five classes.

Basically, two approaches were adopted: a set of hand-crafted rules and machine-learning models. Examples of the rules for extent and *type* are listed in Table 5. In the first rule of the table, the first condition is satisfied when the sequence of two morphemes is a digit followed by a morpheme '월 [wol]' (month), '일 [il]' (day), or '주 [joo]' (week). The various ways of expressing numbers are also considered. The second condition is satisfied when the morpheme next to the extent is '에 [eh]' (at) or '마다 [ma-da]' (every) followed by '번 [beon]' (times) or '회 [hoi]' (times), and there must be no other tags between the two morphemes. If these two conditions are satisfied, then the sequence of morphemes becomes the extent of the *timex3* tag of which *type* is SET. In the fourth rule, the first condition is satisfied when the corresponding morpheme is either '새벽 [sae-byuk]' (dawn), '오전 [o-jeon]' (before noon), '오후 [o-hoo]' (after noon), '아침 [a-chim]' (morning), '점심 [jeom-sim]' (noon), '저녁 [jeo-nyuk]' (evening), '낮 [nat]' (daytime), or '밤 [bam]' (night). The second condition is satisfied when the next morpheme is '경 [gyoung]' (about) or '에 [eh]' (at). If these two conditions are satisfied, then the *type* of the corresponding morpheme becomes TIME. For each morpheme, all the rules are checked in ascending order. When one of the rules is matched, then the remaining rules are skipped for the target morpheme. When no rule is matched, then the *type* of the corresponding morpheme becomes NONE.

Table 5. Examples of the rules for extent and type of *timex3* tags

Type	Conditions
SET	Extent=(digit,월∨일∨주), Next morps=(에∨마다,no other tags, 번∨회)
DURATION	Extent=(digit,세기), Next morps=(동안)
DATE	Extent=(음력,digit,월)
DATE	Extent=(digit,년)
TIME	Extent=(새벽∨오전∨오후∨아침∨점심∨저녁∨낮∨밤), Next morps=(경에)

Table 6. Examples of the rules for value of timex3 tag

Operations	Conditions
Day = digit, Context updated	Type=DATE∨TIME, Surrounding morps=(digit,일)
Month = digit, Context updated	Type=DATE∨TIME, Surrounding morps=(digit,월)
Month = context+1	Type=DATE∨TIME, Surrounding morps=(다음달)
Month = context-digit	Type=DATE∨TIME, Surrounding morps=(digit,개월전)
Year = -digit, Context updated	Type=DATE∨TIME, Surrounding morps=(기원전,digit,년)
Year = context	Type=DATE∨TIME, Surrounding morps=(올해∨이번해)

For machine-learning models, a set of features is defined given a particular window size: unigrams of morphemes, POS tags, NE tags, and morpheme-level features of dependency parsing. The morpheme-level features of dependency parsing are generated by the following approaches in [21]. These features are defined using the raw features generated by the Korean analyzer, which may cause error propagation when the raw features contain some errors. That is, the four extractors (i.e., *timex3* extractor, *event* extractor, *makeinstance* extractor, and *tlink* extractor) work in a cascade, so the errors within the raw features will propagate through the entire process from the *timex3* extraction to the *tlink* extraction. Thus, other than these features, we adopt some complementary features directly generated from the texts, as done in our previous work [1]. Here, we employ the features generated by the Language Independent Feature Generator (LIFE) model [22], where the model takes the texts as input and generates semantic/syntactic features.

For the extent and *type* prediction, a set of 222 rules is defined. To predict other attributes of each predicted *timex3* tag, several sets of rules are defined: 117 rules for *value,* 8 rules for *beginPoint/endPoint,* 10 rules for *mod,* 10 rules for *prd,* 1 rule for *quant,* 1 rule for *freq,* and 1 rule for *calendar.* To make it easy to manipulate the rules, all the rules are written using a predefined meta-language similar to regular expression.

The rules for value and *freq* take the *temporal context* into account. For the sentence "I go there tomorrow", it is hard to predict the *value* of 'tomorrow' without considering the temporal context. It is assumed that the temporal context of each sentence depends on the previous sentence. For each document, the temporal context is initialized with Document Creation Time (DCT), and the context is updated when a normalized *value* appears in a certain condition. Examples of the rules for *value* are described in Table 6. For the first rule in the table, there are two conditions. The first condition is whether the *type* is DATE or TIME. The second condition is whether the tag extent contains a digit and '일 [il]' (day). When the two conditions are satisfied, then the day of *value* is changed to the corresponding digit and the temporal context is updated. Note that, when the temporal unit is updated, the context of higher temporal units is also updated. For example, when the day of *value* is updated, the year and month are also updated. To avoid overwriting value with wrong values from satisfying multiple rules, the rules are listed in ascending order (level) of temporal units. That is, the rules for seconds or minutes are listed before the rules for hours or days. This allows *value* to be changed from lower temporal units to higher units, thereby avoiding overwriting with the wrong *value.*

Because there can be multiple clues for determining value within an extent, all of the rules are checked for each *timex3* tag. More precisely, all the rules of each time unit are checked until there are no matched rules of the time unit. For example, if a particular rule about day is matched, then all the remaining rules about day are skipped. Thereafter, the rules about month and year are checked.

As depicted in Fig. 2, some missing *values* of *timex3* tags are filled by communication with the External Knowledge-Base (EKB). The input of the EKB is the Subject-Predicate-Object (SPO) triples, and the output is either a temporal point or a temporal interval. For example, given the sentence "임진왜란 당시 사망했다" (died during the Japanese invasion of Korea), there will be a *timex3* tag of which text = '임진왜란 당시' (during Japanese invasion of Korea). The *timex3* extractor asks the EKB for information about the triple (S = '임진왜란 [im-jin-we-ran]' (Japanese invasion of Korea), P = '', O = ''), and will get the response in a form of temporal interval (e.g., '1592'-'1598'). With the response, two non-consuming *timex3* tags will be generated, and their *values* will be '1592' and '1598', respectively. Thereafter, they will become *beginPoint* and *endPoint* of the *timex3* tag of which text = '임진왜란 당시'. Finally, the *value* of the *timex3* tag is found to be 'P6Y'. As the input of the EKB is the SPO triple, more complicated query is available. For instance, given the triple (S = '세종대왕' (Sejong the Great), P = '반포하다' (distribute), O = '훈민정음' (Hunminjeongeum Manuscript)), the EKB will provide as the response temporal point '1446'. In this case, the *value* can be directly filled with '1446'. In the next section, we demonstrate how much the communication with the EKB help to improve the performance of the *timex3* extractor.

4. EVENT Extractor and MAKEINSTANCE Extractor

The *event* extractor and *makeinstance* extractor are basically similar to those in previous work. The goal of the *event* extractor is to predict whether each morpheme belongs to the extent of an *event* tag or not, and finds an appropriate *class* of the tag. Similar to the *event* extractor, it predicts the *class* of each morpheme, where there are seven *classes* of *event* tag: OCCURRENCE, PERCEPTION, REPORTING, STATE, I_STATE, I_ACTION, and NONE. The NONE represents that the corresponding morpheme does not belong to the extent. This is essentially a morpheme-level classification over seven classes.

Two approaches are employed: a set of rules and machine-learning models (e.g., CRF and MEM). As the rule-based approach, a set of 27 rules is manually defined. A set of features used in the *timex3* extractor is also used for the machine-learning approach, and the complementary features generated from the LIFE model are also adopted.

We define a set of rules to eliminate some verbal morphemes that often appear within the extents of *event* tags, although they do not carry any meaning. For example, in the sentence, "나는 공부를 하다" (I study), the verb '하 [ha]' (do) has no meaning while the noun '공부 [gong-bu]' (study) has the meaning of the event 'study'. Using such rules makes it possible to avoid generating meaningless *event* tags.

The goal of the *makeinstance* extractor is to generate at least one *makeinstance* tag for each *event* tag, and predict appropriate attribute values. The *makeinstance* tag plays the role of event instance, so it has almost all the attributes of the event, while the *event* tag has an extent and the class attribute. The ExoTime system simply generates one *makeinstance* tag for each *event* tag, because it was observed that there is only one *makeinstance* tag for each *event* tag in most cases. For the attributes, we define several sets of rules: 11 rules for *tense,* 2 rules for *polarity,* 3 rules for *modality,* and 1 rule for *cardinality.* For the attribute *POS,* the POS tags generated by the Korean analyzer is used.

5. TLINK Extractor

The goal of *tlink* extractor is to make a linkage between two tags, and to find appropriate types of links. More precisely, given each pair of tags, it predicts the value of the attribute *relType,* where there are 11 *relTypes:* BEFORE, AFTER, INCLUDES, DURING, DURING₩_INV, SIMULTANEOUS, IDENTITY, BEGINS, ENDS, OVERLAP, and NONE. The NONE represents that there is no linkage between the two argument tags. Thus, it is essentially a classification over 11 classes for each pair of two argument tags.

The *tlink* extractor generates inter-sentence *tlinks* and intra-sentence *tlinks*. For the inter-sentence *tlinks,* MM *tlinks* between adjacent sentences are generated when there is a particular expression at the beginning of a sentence, such as '그 후 [geu-hoo]' (afterward), '그 전 [geu-jeon]' (beforehand), or '그 다음 [geu-da-eum]' (thereafter). For the intra-sentence *tlinks,* two approaches are considered to extract TM *tlinks* and MM *tlinks:* a rule-based approach and a data-driven approach (e.g., SVM and LR). For the rule-based approach, a set of 24 rules is defined. For the data-driven approach, two independent models were trained to predict TM *tlinks* and MM *tlinks,* respectively. We adopted the features used in [1], as described in Table 7. To improve performance, especially to improve recall, we also adopted a bagging method to train the machine-learning models. The reason for improving recall is that greater recall gives results that seem qualitatively better. The bagging method allows the models to achieve better recall with a relatively small loss of precision, which finally results in better performance.

Table 7. Features in the two kinds of classifiers

Type	Features
TM *tlink*	• Surrounding morphemes of the argument tags • Linear order of the argument tags • Attribute *type* of *timex3* tag • Attribute *class* of *event* tag • Attribute *polarity* of *makeinstance* tag • Attribute *tense* of *makeinstance* tag • Whether *tlink* tag is non-consuming tag or not • Does *timex3* tag exist between argument tags? • Does any *event* tag exist between argument tags? • Is the corresponding *event* tag an objective of other *event* tag in dependency tree?
MM *tlink*	• Surrounding morphemes of *event* tags • Linear order of the corresponding *event* tags • Attribute *class* of *event* tags • Attribute *polarity* of *makeinstance* tags • Attribute *tense* of *makeinstance* tags • Does *timex3* tag exist between *event* tags? • Does any *event* tag exist between the corresponding *event* tags?

IV. Experiments

We use Korean TimeBank v1.1 [19] as a dataset. The Korean TimeBank is regularly updated, and the statistics of it are described in Table 8. The dataset was divided into a training dataset, a validation dataset, and a test dataset. The validation dataset was used to optimize parameters of machine-learning methods, and the test dataset was completely unseen before the final evaluation. As shown in the table, only one *makeinstance* tag exists for each event tag in most cases, which follows the assumption of the *makeinstance* extractor.

Table 8. The statistics of Korean TimeBank, where the digits represent the number of corresponding items

Item	Training	Validation	Test
documents	610	208	260
sentences	2,626	543	905
timex3	1,656	377	524
event	7,650	1,440	2,479
makeinstance	7,682	1,448	2,494
tlink	2,663	478	847

For the experiments, we utilized External Knowledge-Base (EKB), namely oracle. The oracle is a set of tuples, each of which takes a form of {S, V, O, and T}, where S is subject, V denotes verb, O means object, and T is a temporal information. For example, {The Declaration of Independence, nil, nil, 1776-07-04} means that the event 'The Declaration of Independence' happened in 1776-07-04. For the tuple representing a period of event (e.g., The Second World War), the term T constitutes of two temporal points (e.g., '1939-09-01 and 1945-09-02'). Using EKB can be very helpful to develop some real-world applications (e.g., Question-Answer systems). Without the information about when 'the Second World War' happened, it will be impossible to answer the question "Who was the king of Korea during the Second World War?". In the following experiments, we utilized only tens of tuple instances, and we continuously extend it.

1. TIMEX3 Extraction

For the extent and *type* prediction, only the exactly predicted extents and *types* are regarded as correct, and the results are summarized in Table 9. There, MEM was trained only with the features generated from raw features, and MEM_L was trained with both the features and LIFE features. The performance is measured in a sequential manner, so the performance generally decreases from the top to bottom of the table.

Table 9. Timex3 prediction results, where points represent *beginPoint/endPoint,* and *cal* represents calendar

Attributes	Performances			
	Combination	Precision	Recall	F1 score
extent	Rules	67.87	67.11	67.49
	Rules (EKB)	69.86	76.69	73.12
	MEM	30.54	17.37	22.14
	CRF	75.54	53.63	62.72
	MEM, Rules (EKB)	67.27	76.88	71.75
	CRF, Rules (EKB)	69.05	**78.01**	73.26
	MEML	33.77	19.85	25.00
	CRFL	**82.44**	61.83	70.67
	CRFL, Rules (EKB)	69.41	77.63	**73.29**
type	Rules	66.35	65.60	65.97
	Rules (EKB)	68.49	75.19	71.68
	MEM	29.53	16.79	21.41
	CRF	73.66	52.29	61.16
	MEM, Rules (EKB)	65.79	75.19	70.18
	CRF, Rules (EKB)	67.55	**76.32**	71.67
	MEML	33.12	19.47	24.52
	CRFL	**81.17**	60.88	69.57
	CRFL, Rules (EKB)	67.89	75.94	**71.69**
value	Rules	60.17	67.29	63.53
points	Rules	59.16	66.17	62.47
freq	Rules	59.16	66.17	62.47
mod	Rules	57.82	64.66	61.05
cal	Rules	57.82	64.66	61.05
prd	Rules	57.82	64.66	61.05
quant	Rules	57.82	64.66	61.05

The CRF++ library [25] and MEM toolkit [26] were used. The optimal parameter settings are found by a grid search with the validation set. The optimal setting for CRF is as follows: L1-regularization, c = 0.6, and f = 1. The MEM shows its best performance without Gaussian prior smoothing. Both models give generally better performance when window size is [-2, +2]. The parameter setting for O-LIFE is as follows: a = 0.1, b_t = 0.1, g = 0.1, b_c = 0.1, w^p = (1), C = 10, T = 3, $t\ 1_F$ = 0.1, $t\ 2_F$ = 0.5, $t\ 3_F$ = 0.1, $t\ 4_F$ = 0.5, $t\ 1_B$ = 0.1, $t\ 2_B$ = 0.05, $t\ 3_B$ = 0.1, $t\ 4_B$ = 0.5, and the number of iterations for estimation is 1000.

As shown in the table, the rule-based approach gives quite good performance, which is consistent with the reports of TempEval-3. In the TempEval-3, for the task of extraction of *timex3,* the best performance was 77.61% (F1 measure) achieved by HeidelTime-t [23] which is a rule-based system. The rule-based approach seems to be powerful for the task of extraction of *timex3,* while the combination of the rules and the machine-learning models gives comparative performance to the rule-based approach. The reason for this might be that the same temporal value can be expressed in various forms in texts. For example, '2010-05-05T15:00' can be expressed by '2010년 5월5일 15시', '2010월 5월5일 오후 3시', or '2010/5/5 15시'. Such various ways of representing same temporal value could not be captured by the machine-learning models, but it may be improved with a huge amount of training instances in future. It is also observed that using the External Knowledge-Base (EKB) improves the performance. For example, the rules with EKB increases the performance about 6 percent of F1 score.

When the CRF is trained using the LIFE features, namely CRF $_L$, performance is better than when using only the rules. This seems to be due to the raw features of Korean (e.g., POS tagger): being unstable, and that the LIFE features complement these unstable features by capturing syntactic/semantic patterns that are inherent in the given documents. The combination of CRF $_L$ and rules gives the best performance, where the rules contribute to increase recall with only a small loss of precision. With the best combination, the other remaining attributes are predicted using the rules.

To summarize, the rule-based approach has a huge impact on the performance of the task of *timex3* extraction, and using the LIFE features contributes to the improvement of performance. As a future research direction, intensive rule engineering, based on linguistic knowledge and observations, is necessary to capture the complex patterns of temporal expressions. It is also necessary to find a way to wisely combine the machine-learning model and the rules.

2. EVENT Extraction

The results of *event* prediction are summarized in Table 10.

Table 10. *Event* prediction results

Attributes	Performances			
	Combination	Precision	Recall	F1 score
extent	Rules	85.55	33.76	48.42
	MEM	38.55	16.94	23.54
	CRF	78.85	59.08	67.55
	MEM, Rules	44.75	34.82	39.16
	CRF, Rules	81.03	64.25	71.67
	MEML	41.00	18.24	25.24
	CRFL	**88.19**	82.13	**85.05**
	CRFL, Rules	85.73	**83.78**	84.74
class	Rules	73.46	28.99	41.58
	MEM	37.07	16.30	22.64
	CRF	68.16	51.07	58.39
	MEM, Rules	38.20	29.72	33.43
	CRF, Rules	69.20	54.87	61.21
	MEML	40.18	17.87	24.74
	CRFL	**82.54**	**76.87**	**79.61**
	CRFL, Rules	77.91	76.14	77.01

The performance is measured in a sequential manner, so the performance generally decreases from the top to bottom of the table. By a grid search, the optimal parameter settings are found to be the same as that of the *timex3* extractor.

As shown in the table, for the task of *event* prediction, the rule-based approach is not powerful, while the data-driven approach seems powerful. This is consistent with the reports of TempEval-3, of which it is reported that, for the task of extraction of *event* and *makeinstance* tags, the best performance was 81.05% (F1 measure) achieved by ATT-1 [24], which took a data-driven approach. The combination of the rules and the machine-learning model (e.g., CRF, MEM) shows better performance than using only the machine-learning models. For example, the combination of CRF and rules gives better performance than using only CRF. It seems that the rules and CRF complement each other to increase recall with only a small loss of precision. The CRF trained using the LIFE features, namely CRF $_L$, gives the best performance. This seems due to the raw features of Korean (e.g., POS tagger) being unstable, and the LIFE features complement these unstable features by capturing syntactic/semantic patterns that are inherent in the given documents. The combination of the CRF $_L$ and rules gives worse performance than using only CRF $_L$, which implies that using the LIFE features is enough to capture arbitrary patterns.

The big difference between *timex3* and *event* is that the *timex3* tag has relatively more complex attributes to normalize. For instance, in order to predict the attribute *value* of *timex3* tag, it will be necessary to define some rules by considering linguistic patterns. This makes difficult to predict the extent of *timex3* tags. On the other hand, the *event* tag does not have that kind of attributes, so it is relatively easy to train the machine-learning models to grasp the patterns.

To summarize, the data-driven approach (especially, CRF) has a huge impact on the performance of the task of *event* extraction, and using the LIFE features contributes to the improvement of performance. As a future research direction, intensive feature engineering based on a linguistic knowledge and observations is necessary to capture the complex patterns to distinguish whether a particular noun expression is an event expression or not. It is also necessary to find a way to wisely combine the machine-learning model and the rules.

3. MAKEINSTANCE Extraction

All the attributes of *makeinstance* tag are predicted through hand-crafted rules. The performance of *makeinstance* prediction is obtained using the predicted *event* tags generated by the CRF $_L$ which is the best approach for event prediction. It is summarized in Table 11, where the measurement of the attribute *POS* is excluded because it is simply the result from the Korean analyzer.

Table 11. *Makeinstance* prediction results

Attributes	Performances		
	Precision	Recall	F1 score
eventID	88.15	81.59	84.74
polarity	93.20	76.05	83.75
tense	68.13	51.81	58.86
modality	99.92	51.77	68.20
cardinality	99.85	51.73	68.17

4. TLINK Extraction

For the linkage and *relType* prediction, the rule-based approach and the data-driven approach (e.g., SVM and LR) were employed. In terms of the data-driven approach, two machine-learning models were trained independently to predict TM *tlinks* and MM *tlinks*, respectively. The TM *tlink* is the *tlink* between *timex3* and *makeinstance* tags, while

MM *tlink* is the *tlink* between two *makeinstance* tags. By performing a grid search with the validation set, it was found that Support Vector Machine (SVM) of C-SVC type with Radial Basis Function (RBF), gives the best performance when Γ of kernel function is 1/number of features. It was also found that the L1-regularized Logistic Regression (LR) with $C = 1$ gives the best performance. It is observed that both models give better performance when window size is [-1, +1].

Fig. 3. The ratio of data instances of NONE relType within the training/validation dataset

The *tlink* prediction is essentially a classification of the attribute *relType* given two argument tags, where the attribute value NONE denotes that there is no temporal relation between the two argument tags. It was observed that the number of data instances having NONE was much larger than the number of data instances having other attribute values. The ratio of the data instances of NONE *relType* within the training/validation dataset is depicted in Fig. 3. The blue color indicates the ratio of NONE *relType,* and the other colors indicate the ratio of the remaining *relType* values. The reason for such imbalance is that almost all pairs of argument tags have no temporal relation. This causes low recall when using SVM, which in turn causes no benefit from the combination of rule-based approach and the data-driven approach. There are two ways to address this situation: up-sampling and down-sampling. As shown in Fig. 3, there are some *relType* values that scarcely occur or do not occur, such as IDENTITY. Due to such extreme ratios, up-sampling is not a good choice. In terms of down-sampling, the data instances of NONE can be down-sampled given a particular ratio of NONE. However, using only the down-sampling method was not very helpful, because it increases recall but with a large loss of precision. Thus, the bagging method was employed with the down-sampling. The data instances of NONE are randomly down-sampled into each bag by putting each NONE data instance into the bags given a particular ratio *p*, while all the other data instances are copied into each bag. This finally gave high recall with a relatively small loss of precision. Through a grid search, it was found that the most effective number of bags is five, and the ratio *p* for the TM *tlinks* is 0.4 and *p* for the MM *tlinks* is 2.7. In terms of the LR, it exhibits low precision, which is the opposite from the SVM. By a grid search, it was observed that applying the bagging to the LR does not contribute to performance improvement, so a somewhat different strategy was adopted. That is, the bagging for the LR was performed by putting each NONE data instance into the bags without replacement, while all the other data instances were copied into each bag. This yields the result that the proportions of the NONE data instances versus all the other data instances are 0.53 and 3.47, for the TM *tlinks* and the MM *tlinks,* respectively.

Table 12. Tlink prediction results given correct *timex3, event,* and *makeinstance* tags

Attributes	Performances			
	Combination	Precision	Recall	F1 score
linkage	Rules	43.86	43.50	43.68
	SVM	**65.08**	40.31	49.78
	SVM, Rules	42.70	50.12	46.11
	LR	21.03	37.23	26.88
	LR, Rules	25.67	59.10	35.79
	SVM(B)	48.48	52.72	**50.51**
	LR(B)	21.47	43.14	28.67
	SVM(B), Rules	39.92	**59.69**	47.84
	LR(B), Rules	25.67	59.10	35.79
relType	Rules	42.91	42.55	42.73
	SVM	**64.50**	39.95	49.34
	SVM, Rules	41.89	49.17	45.24
	LR	19.89	35.22	25.43
	LR, Rules	24.64	56.74	34.36
	SVM(B)	47.83	52.01	**49.83**
	LR(B)	20.47	41.13	27.34
	SVM(B), Rules	39.05	**58.39**	46.80
	LR(B), Rules	24.64	56.74	34.36

There are two cases of tlink prediction: (1) *tlink* prediction given correct other tags, and (2) *tlink* prediction given predicted other tags. The results of the first case are summarized in Table 12, where SVM(B) indicates the SVM with bagging. The performance is measured in a sequential manner. As shown in the table, SVM(B) gives the best performance. The LIFE features were not used for *tlink* extraction because it was observed that the LIFE features do not contribute to the performance of *tlink* extraction. This appears to be because the LIFE features represent only the syntactic/semantic patterns of the given terms, but not the arbitrary relational patterns between the two arguments.

The results of the second case (i.e., *tlink* prediction given predicted other tags) were obtained using the best combination which is the SVM(B), and are shown in Table 13. Note that the *tlink* tags are predicted based on the other tags (e.g., *timex3, event, makeinstance*) that are obtained using the LIFE features. To measure the impact of the LIFE features, a second experiment was conducted, in which the predicted tags were used without using the LIFE features, and the results are shown in Table 14. That is, the *timex3* tags are obtained using the rules, while the *event* tags are obtained using the combination of rules and CRF. As shown in Table 13 and Table 14, using the LIFE features increases the F1 score by about 1%.

Table 13. Tlink prediction results of the SVM(B), given *timex3, event,* and *makeinstance* tags which are predicted using LIFE features

Attributes	Performances		
	Precision	Recall	F1 score
linkage	31.02	29.55	30.27
relType	30.89	29.43	30.15

Table 14. *Tlink* prediction results of the SVM(B), given the tags predicted without using the LIFE features

Attributes	Performances		
	Precision	Recall	F1 score
linkage	31.49	27.54	29.38
relType	31.35	27.42	29.26

To summarize, the data-driven approach (especially, SVM with bagging) has huge a impact on the performance of the task of *tlink* extraction. As a future research direction, intensive feature engineering is necessary to capture the complex patterns about the relations between events. It will be also necessary to explore some other resources (e.g., KorLex) and to build better feature generators, and to find a way to wisely combine the machine-learning model and the rules.

V. Discussion

For the *event* extraction, the handcrafted rules tend to have higher precision, while the machine-learning models have better coverage. The combination of them increased the total performances. For the *timex3* extraction, using only rules gave quite good performances with high precision and recall. The reason is that the temporal expressions have very complex linguistic patterns, so we needed to design the rules very carefully about the *timex3* attributes (e.g., value). Meanwhile, in the *timex3* extraction and tlink extraction, we observed that combination of rules and machine-learning models improved the overall performances. The reason is that the machine-learning models captured the patterns which were not defined by rules.

The *tlink* extraction in this paper was evaluated in the same manner as the TempEval campaigns. The only difference is that we showed two different evaluation results about the *tlink* prediction with the predicted other tags. The reason is that we want to show that using LIFE features help to improve the overall performances.

The temporal information can be represented in various ways, and the various ways will probably different with different languages. In other words, given a specific language, it is necessary to study the linguistic knowledge about the language and take many observations about the linguistic patterns in the real-world texts. Following this process helps to define/design better rules or features, and eventually will make the overall system to be better. Thus, based on linguistic knowledge and observations, we need to continuously find better features and rules, and also better way to combine them.

VI. Conclusions

There are mainly three approaches for temporal information extraction: the rule-based approach, the data-driven approach, and the hybrid approach. As the temporal information has a complex structure, it seems necessary to take the hybrid approach to extract such complex information. However, it was observed that each sub-task has a distinct tendency to give better performance with a particular approach. In terms of the task of *timex3* extraction, it can be seen that it mainly relies on the rule-based approach, because the performance from using only rules is similar to or better than, the performance of the combination of rules and machine-learning models. The reason for this might be that the same temporal value can be expressed in various forms in texts, so the temporal patterns are too complex to predict without rules defined based on linguistic knowledge and observations. We also used an External Knowledge-Base (EKB) for the *timex3* extraction, in order to improve performance of value prediction. It was observed that the task of event extraction and the task of tlink extraction mainly depend on the data-driven approach. We employed the bagging method for the tlink extraction, and gained high recall with a relatively small loss of precision. Although we used the LIFE features as complementary features to compensate for errors within the raw features, there are many other feature generation methods (e.g., tree-kernel, word embedding). Thus, it will be necessary to investigate alternative feature generation methods in the future.

REFERENCES

[1] Y. S. Jeong, Z. M. Kim, H. W. Do, C. G. Lim, and H. J. Choi, "Temporal Information Extraction from Korean Texts," In *Proceedings of the 19th Conference on Computational Language Learning,* Beijing, China, pp. 279-88, 2015.

[2] J. Pustejovsky, J. Castano, R. Ingria, R. Sauri, R. Gaizauskas, A. Setzer, and G. Katz, "TimeML: Robust Specication of Event and Temporal Expressions in Text," In *New Directions in Question Answering,* Stanford, USA, pp. 28-34, 2003.

[3] T. Caselli, V. B. Lenzi, R. Sprugnoli, E. Pianta, and I. Prodanof, "Annotating Events, Temporal Expressions and Relations in Italian: the It-TimeML Experience for the Ita-TimeBank," In *Proceedings of the Fifth Law Workshop,* Partland, Oregon, pp. 143-51, 2011.

[4] S. Im, H. You, H. Jang, S. Nam, and H. Shin, "KTimeML: Specication of Temporal and Event Expressions in Korean Text," In *Proceedings of the 7th Workshop on Asian Language Resources*, Suntec, Singapore, pp. 115-22, 2009.

[5] M. Verhagen, R. J. Gaizauskas, F. Schilder, M. Hepple, J. Moszkowicz, and J. Pustejovsky, "The TempEval Challenge: Identifying Temporal Relations in Text," *Language Resources and Evaluation,* Vol. 43, No. 2, pp. 161-79, 2009.

[6] M. Verhagen, R. Sauri, T. Caselli, and J. Pustejovsky, "SemEval-2010 task 13: TempEval-2," In *Proceedings of the Fifth International Workshop on Semantic Evaluation,* Uppsala, Sweden, pp. 57-62, 2010.

[7] N. UzZaman, H. Llorens, L. Derczynski, M. Verhagen, J. Allen, and J. Pustejovsky, "SemEval-2013 Task 1: TEMPEVAL-3: Evaluating Time Expressions, Events, and Temporal Relations," In *Proceedings of the Seventh International Workshop on Semantic Evaluation*, Atlanta, Georgia, USA, pp. 1-9, 2013.

[8] A. Berglund, R. Johansson, and P. Nugues, "A Machine Learning Approach to Extract Temporal Information from Texts in Swedish and Generate Animated 3D Scenes," In *Proceedings of the 11st Conference of the European Chapter of the Association for Computational Linguistics,* Trento, Italy, pp. 385-92, 2006.

[9] N. Chambers, S. Wang, and D. Jurafsky, "Classifying Temporal Relations Between Events," In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, Prague, Czech Republic, pp. 173-6, 2007.

[10] N. UzZaman and J. Allen, "Event and Temporal Expression Extraction from Raw Text: First Step towards a Temporally Aware System," *International Journal of Semantic Computing,* Vol. 4, No. 4, pp. 487-508, 2010.

[11] J. Strotgen, M. Gertz, and P. Popov, "Extraction and Exploration of Spatio-Temporal Information in Documents," In *Proceedings of the 6th Workshop on Geographic Information Retrieval Article,* Zurich, Switzerland, pp. 698-706, 2010.

[12] B. Tang, Y. Wu, M. Jiang, Y. Chen, J. C. Denny, and H. Xu, "A Hybrid System for Temporal Information Extraction from Clinical Text," *Journal of the American Medical Informatics Association,* Vol. 20, No. 5, pp. 828-35, 2013.

[13] T. Cassidy, "Temporal Information Extraction and Knowledge Base Population," PhD thesis, The City University of New York, 2014.

[14] N. Chambers, T. Cassidy, B. McDowell, and S. Bethard, "Dense Event Ordering with a Multi-Pass Architecture," *Transactions of the Association for Computational Linguistics,* Vol. 2, pp. 273-84, 2014.

[15] S. B. Jang, J. Baldwin, and I. Mani, "Automatic TIMEX2 Tagging of Korean News," *ACM Transactions on Asian Language Information Processing*, Vol. 3, No. 1, pp. 51-65, 2004.

[16] C. N. Seon, S. Kang, and J. Seo, "Automatic Recognition and Normalization System of Korean Time Expression Using the Individual Time Units," *Cognitive Science*, Vol. 21, No. 4, pp. 447-58, 2010.

[17] Y. Kim and J. Choi, "Recognizing Temporal Information in Korean Clinical Narratives through Text Normalization," *Healthcare Information Research,* Vol.

17, No. 3, pp. 150-5, 2011.
[18] G. Angeli and J. Uszkoreit, "Language-Independent Discriminative Parsing of Temporal Expressions," In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics,* Soa, Bulgaria, 2013.
[19] Y. S. Jeong, W. T. Joo, H. W. Do, C. G. Lim, K. S. Choi, and H. J. Choi, "Korean TimeML and Korean TimeBank," In *Proceedings of the 10th edition of the Language Resources and Evaluation Conference, Portoroz,* Slovenia, pp. 356-9, 2016.
[20] S. Lim, C. K. Lee, J. Hur, and M. G. Jang, "Syntax Analysis of Enumeration type and Parallel Type Using Maximum Entropy Model," In *Proceedings of the Korea Human Computer Interaction Conference,* 1240-5, 2006.
[21] E. F. T. K. Sang, and S. Buchholz, "Introduction to the CoNLL-2000 Shared Task: Chunking," In *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning*, Lisbon, Portugal, pp. 127-32, 2000.
[22] Y. S. Jeong and H. J. Choi, "Language Independent Feature Extractor," In *Proceedings of the Twenty-Ninth AAAI Conference on Articial Intelligence,* Texas, USA, pp. 4170-1, 2015.
[23] J. Strotgen, and M. Gertz, "HeidelTime: High Quality Rule-based Extraction and Normalization of Temporal Expressions," In *Proceedings of the Fifth International Workshop on Semantic Evaluation,* Uppsala, Sweden, pp. 321-4, 2010.
[24] H. Jung and A. Stent, "ATT1: Temporal Annotation Using Big Windows and Rich Syntactic and Semantic Features," In *Proceedings of the Seventh International Workshop on Semantic Evaluation,* Atlanta, Georgia, pp. 20-4, 2013.
[25] CRF++ library, http://crfpp.googlecode.com/svn/trunk */doc/index.html*
[26] MEM toolkit, http://homepages.inf.ed.ac.uk/lzhang10 */maxent_toolkit*

Authors

Young-Seob Jeong received a BS in Computer Science from Hanyang University, Korea, in 2012, an MSc in Computer Science from KAIST, Korea, and in 2016, a PhD in School of Computing from KAIST, Korea. He joined the faculty of the Department of Big Data Engineering at Soonchunhyang University, Asan city, Korea, in 2017. His current research topics are text mining, information extraction, action recognition, and dialog systems, where his fovorite techniques are topic modeling and deep learning.

Chae-Gyun Lim is currently a PhD candidate in the School of Computing at KAIST, Korea. In 2011, he received a BS in Medical Computer Science from Eulji University, Korea. Between 2011 and 2013, he worked as a research assistant in the Dept. of Computer Science at KAIST, Korea, and in 2015, a MSc in the Dept. of Computer Engineering at Kyung Hee University, Korea. His research interests include temporal information extraction, topic modeling, big data analysis and bioinformatics.

Ho-Jin Choi is currently an associate professor in the School of Computing at KAIST. In 1982, he received a B.S. in Computer Engineering from Seoul National University, Rep. of Korea. In 1985, he obtained an MSc in Computing Software and Systems Design from Newcastle University, UK, and in 1995, a Ph.D. in Artificial Intelligence from Imperial College, London, UK. Currently, he serves as a member of the board of directors for the Software Engineering Society of Korea, the Computational Intelligence Society of Korea, and the Korean Society of Medical Informatics. His current research interests include artificial intelligence, data mining, software engineering, and biomedical informatics.