

# Temporal Information Extraction from Korean Texts

Young-Seob Jeong, Zae Myung Kim, Hyun-Woo Do, Chae-Gyun Lim, and Ho-Jin Choi

School of Computing

Korea Advanced Institute of Science and Technology

291 Daehak-ro, Yuseong-gu, Daejeon 305-701, South Korea

{pinode, zaemyung, realstorm103, rayote, hojinc}@kaist.ac.kr

## Abstract

As documents tend to contain temporal information, extracting such information is attracting much research interests recently. In this paper, we propose a hybrid method that combines machine-learning models and hand-crafted rules for the task of extracting temporal information from unstructured Korean texts. We address Korean-specific research issues and propose a new probabilistic model to generate complementary features. The performance of our approach is demonstrated by experiments on the TempEval-2 dataset, and the Korean TimeBank dataset which we built for this study.

## 1 Introduction

Due to the increasing number of unstructured documents available on the Web and from other sources, developing techniques that automatically extract knowledge from the documents has been of paramount importance. Among many aspects of extracting knowledge from documents, the extraction of temporal information is recently drawing much attention, since the documents usually incorporate temporal information that is useful for further applications such as Information Retrieval (IR) and Question Answering (QA) systems. Given a simple question, “who was the president of the U.S. 8 years ago?”, for example, a QA system may have a difficulty in finding the right answer without the correct temporal information about when the question is posed and what ‘8 years ago’ refers to.

There have been many studies for temporal information extraction, but most of them are applicable only to their target languages. The main reason for this limitation is that some parts of temporal information are difficult to predict without

the use of language-specific processing. For example, the normalized value ‘1983-03-08’ can be represented by ‘March 8, 1983’ in English, while it can be represented by ‘1983년삼월8일’ in Korean. The order of date representation in Korean is usually different from that of English, and the digit expression in Korean is more complex than that of English. This implies that it is necessary to investigate language-specific difficulties for developing a method to extract temporal information.

In this paper, we propose a method for temporal information extraction from Korean texts. The contributions of this paper are as follows: we (1) show how the Korean-specific issues (e.g., morpheme-level tagging, various ways of digit expression, uses of lunar calendar, and so on) are addressed, (2) propose a hybrid method that combines a set of hand-crafted rules and machine-learning models, (3) propose a data-driven probabilistic model to generate complementary features, and (4) create a new dataset, the Korean TimeBank, that consists of more than 3,700 manually annotated sentences.

The rest of the paper is organized as follows. Section 2 describes the background of the research. Section 3 presents the details of the proposed method, the Korean TimeBank dataset, and how we apply the probabilistic model for generating features. Section 4 shows experimental results, and Section 5 concludes the paper.

## 2 Background

TempEval is a series of shared tasks for temporal information extraction (Verhagen et al., 2009; Verhagen et al., 2010; UzZaman et al., 2013). There have been many studies related to the shared tasks (Chambers et al., 2007; Yoshikawa et al., 2009; UzZaman and Allen, 2010; Ling and Weld, 2010; Mirroshandel and Ghassem-Sani, 2012; Bethard, 2013b), which are based on the Time Mark-up Language (TimeML) (Pustejovsky et al., 2003).

The shared tasks can be summarized into three extraction tasks: (1) extraction of *timex3* tags, (2) extraction of *event* and *makeinstance* tags, and (3) extraction of *mlink* tags. The *timex3* tag is associated with expressions of temporal information such as ‘May 1973’ and ‘today’. The *event* tag and *makeinstance* tag represent some eventual expressions which can be related to temporal information. The *makeinstance* tag is an instance of the *event* tag. For example, the sentence, “I go to school on Mondays and Tuesdays”, contains one *event* tag on ‘go’ and two *makeinstance* tags as the action ‘go’ occurs twice on Mondays and Tuesdays. The *mlink* tag represents a linkage between two tags. The *mlink* can be a linkage between two *timex3* tags (TT *mlink*), two *makeinstance* tags (MM *mlink*), a *timex3* tag and a *makeinstance* tag (TM *mlink*), or Document Creation Time and a *makeinstance* tag (DM *mlink*). Note that the *mlink* takes *makeinstance* tags as arguments, but not the *event* tags, as the *event* tags are merely templates for them. For the above sentence, there will be two TM *mlinks*: go-Tuesdays and go-Mondays. The TT *mlink* is assumed to be easy to extract, so TempEval does not incorporate the TT *mlink* into the task of extracting *mlink* tags.

Among many related studies, there are several leading ones. HeidelTime is proposed for extraction of *timex3* tags (Strotgen and Gertz, 2010). It strongly depends on hand-crafted rules, and showed the best performance in TempEval-2. Llorens et al. (2010) proposed TIPSem for all of the three extraction tasks. It employs Conditional Random Fields (CRF) (Lafferty et al., 2001) for capturing patterns of texts, and defines a set of hand-crafted rules for determining several attributes of the tags. ClearTK is another work proposed for all three extraction tasks (Bethard, 2013a); it utilizes machine-learning models such as Support Vector Machines (SVM) (Boser et al., 1992; Cortes and Vapnik, 1995) and Logistic Regressions (LR), and shows the best performance in TempEval-3.

Although the existing approaches show good results, most of them are applicable only to their target languages. The first reason is that there are several attributes which are difficult to predict without the use of language-specific processing. For instance, the attribute *value* of *timex3* tag has a normalized form of time (e.g., 1999-04-12) following ISO-8601. This is not accurately

predictable by relying solely on data-driven approaches. The second reason is that they depend on some language-specific resources (e.g., WordNet) or tools. Unless the same quality of resources or tools is achieved for other languages, the existing works would not be available to the other languages. To alleviate this limitation, a language independent parser for extracting *timex3* tags is proposed (Angeli and Uszkoreit, 2013). Its portability is demonstrated by experiments with TempEval-2 dataset of six languages: English, Spanish, Italian, Chinese, Korean, and French. However, the performance in English and Spanish datasets are about twice as high as the other languages, since the method highly depends on the feature definition and language-specific preprocessing (e.g., morphological analysis). This implies that it is necessary to address language-specific difficulties in order to achieve high performance.

Korean language has many subtle rules and exceptions on word spacing. Korean is an agglutinative language, where verbs are formed by attaching various endings to the stem. There are usually multiple morphemes for each token, and empty elements appear very often because subjects and objects are dropped to avoid duplication. Temporal expressions often take the lunar calendar representation as a tradition. Moreover, the same temporal information can have various forms due to a complex system of digit expression. For instance, a digit 30 can be represented as ‘30’, ‘삼십 [sam-sib]’, or ‘서른 [seo-run]’. Most of these issues stem from the Chinese language, as a large number of Chinese words and letters have become an integral part of Korean vocabulary due to historical contact. All these issues hinder the performance of the existing approaches when applied to Korean documents.

In this paper, we show how these issues are addressed in our Korean-specific hybrid method. To the best of our knowledge, this is the first Korean-specific study which addresses all of the three extraction tasks.

### 3 Proposed Method

#### 3.1 Korean TimeBank

Although there is a Korean dataset provided by TempEval-2, we chose not to use it because it is small in size and has many annotation errors. In TempEval-2 Korean dataset, there are missing values of *timex3* tags, and multiple tags that must be

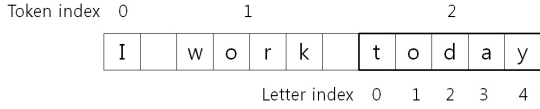


Figure 1: Example of extent representation by token indices and letter indices.

merged into one. Please refer to the examples in the 11th sentence of the 2nd training document of the TempEval-2 Korean dataset. Thus, we constructed a new dataset called Korean TimeBank.

The new dataset is based on TimeML but with several differences. The tags of the new dataset are represented using a stand-off scheme, keeping the original sentences unharmed. As there are often multiple morphemes within each token in Korean, the tags are annotated in letter-level. The letter-level annotation allows multiple annotations to appear within a single token. This also makes the dataset independent of morphological analysis, so it is not required to update the dataset when the morphological analyzer is updated. To enable the letter-level annotation, we introduce several attributes for *timex3* tag and *event* tag: *text*, *begin*, *end*, *e\_begin*, and *e\_end*. The attributes *e\_begin* and *e\_end* indicate token indices, while *begin* and *end* indicate letter indices of the extent. The attribute *text* contains the string of the extent. For example, the sentence, “I work today” in Fig. 1, contains one *timex3* tag whose *text* is ‘today’, where *e\_begin*=2, *e\_end*=2, *begin*=0, and *end*=4.

Since temporal expressions following the lunar calendar representation appear often in Korean, we add an attribute *calendar*. The value of the *calendar* can be LUNAR or other types of calendar, and its default value is GREGORIAN when it is not explicitly clarified. We also add two values for the attribute *mod* of *timex3* tag: START\_OR\_MID and MID\_OR\_END, as these expressions appear often in Korean. For example, ‘초중반[cho-joong-ban]’ represents beginning or middle phase of a period, and ‘중후반[joong-hoo-ban]’ represents middle or ending phase of a period.

The source of the Korean TimeBank includes Wikipedia documents and hundreds of manually generated question-answer pairs. The domains of the Wikipedia documents are personage, music, university, and history. The documents are annotated by two trained annotators and a supervisor, all majoring in computer science. The annotated tags of each document is saved in an XML file.

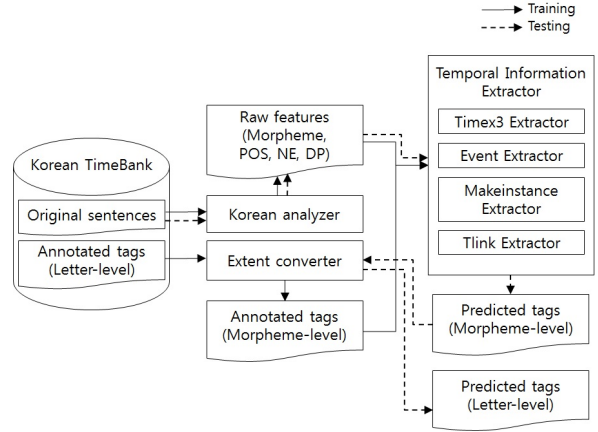


Figure 2: Overall process of temporal information extraction from Korean texts.

### 3.2 Temporal Information Extraction from Korean Texts

Our proposed method addresses all of the three tasks: (1) extraction of *timex3* tags, (2) extraction of *event* and *makeinstance* tags, and (3) extraction of *tlink* tags. The proposed method also extracts additional attributes of *timex3* tag, such as *freq*, *beginPoint*, *endPoint*, *mod*, and *calendar*. The overall process is depicted in Fig. 2, where the solid line represents training process and the dotted line represents testing process. The Korean analyzer at the center of the figure takes Korean texts as an input and generates several raw features as an output, such as results of morphological analysis, Part-Of-Speech (POS) tags, Named-Entity (NE) tags, and results of dependency parsing (Lim et al., 2006). The number of possible POS tags is 45, which follows the definition of Sejong Treebank<sup>1</sup>. The number of possible NE tags is 178, where each of them belongs to one of 15 super NE tags.

The generated raw features are used to define a set of features for machine-learning models and a set of hand-crafted rules. The rules are designed by examining the training dataset and the errors that the proposed method generates with the validation dataset. We employ several machine-learning methods that have shown the best performance in the TempEval shared tasks, such as Maximum Entropy Model (MEM), Support Vector Machine (SVM), Conditional Random Fields (CRF), and Logistic Regression (LR).

Fig. 2 introduces Temporal Information Extractor (TIE) which consists of four sub-extractors:

<sup>1</sup>Korean Language Institute, <http://www.sejong.or.kr>

*timex3* extractor, *event* extractor, *makeinstance* extractor, and *mlink* extractor. The *timex3* extractor and the *event* extractor work independently, and the *makeinstance* extractor uses the predicted *event* tags. The *mlink* extractor makes use of the predicted *makeinstance* tags and predicted *timex3* tags. Thus, the performance of *timex3* extractor and *event* extractor will strongly influence the performance of *makeinstance* extractor and *mlink* extractor. These four sub-extractors as a whole give predicted tags as an output, where the tags are represented in morpheme-level. The extent converter at the center of the figure changes the morpheme-level tags into letter-level and vice versa by checking ASCII values of each letter and each morpheme. In training process, the annotated tags of Korean TimeBank are converted into morpheme-level through the extent converter, and used to train the TIE.

### 3.2.1 Timex3 extractor

The goal of *timex3* extractor is to predict whether each morpheme belongs to the extent of a *timex3* tag or not, and finds appropriate attributes of the tag. There are five *types* of *timex3* tag: DATE, TIME, DURATION, SET, and NONE. The NONE represents that the corresponding morpheme does not belong to the extent of a *timex3* tag, and the other four types follow the same definition of TimeML. This is essentially a morpheme-level classification over 5 classes.

We basically take two approaches: a set of 100 hand-crafted rules and machine-learning models. Examples of the rules for extent and *type* are listed in Table 1. In the second 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 digit expressions 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 *timex3* tag whose *type* is SET. If one of the rules is satisfied, then the remaining rules are skipped for the target morpheme.

We compare two machine-learning models, CRF and MEM, for *timex3* tag by experiments. We defined a set of features based on the raw features

Table 1: Examples of the rules for extent and *type* of *timex3* tags.

Type	Conditions
DATE	Extent=(digit, 년)
SET	Extent=(digit, 월∨일∨주), Next morps= (에∨마다,no other tags, 번∨회)

Table 2: Examples of the rules for *value* of *timex3* tag.

Operations	Conditions
Month=digit Context updated	Type=DATE∨TIME Surrounding morps= (digit, 월)
Year=context	Type=DATE∨TIME Surrounding morps= (올해∨이번해)

obtained from the Korean analyzer. The defined features include morphemes, POS tags, NE tags, morpheme-level features of dependency parsing, given a particular window size. The morpheme-level features of dependency parsing are generated by following approaches in Sang and Buchholz (2000).

To predict other attributes of each predicted *timex3* tag, we also define sets of rules: 112 rules for *value*, 7 rules for *beginPoint/endPoint*, 9 rules for *freq*, 10 rules for *mod*, and 1 rule for *calendar*. Especially, the rules for *value* and *freq* take a temporal context into account. For instance, the sentence “We go there tomorrow”, makes it hard to predict *value* of ‘tomorrow’ without considering the temporal context. We assume 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 are described in Table 2. If the first rule in the table is satisfied, then the month of *value* is changed to the corresponding digit and the temporal context is updated.

As there can be multiple clues for determining *value* within an extent, all of the rules are checked for each *timex3* tag. To avoid overwriting *value* by multiple satisfied rules, the rules are listed in ascending order of temporal unit. That is, the rules

for seconds or minutes are listed before the rules for hours or days. This allows *value* to be changed from smaller temporal unit to bigger unit, thereby avoiding overwriting wrong *value*. The rules for different attributes are listed in separate files, and are written in a systematic way similar to regular expressions. Such format enables rules to be easily manipulated.

### 3.2.2 Event extractor

The goal of *event* extractor is to predict whether each morpheme belongs to the extent of an *event* tag or not, and finds appropriate *class* of the tag. There are 7 *classes* of *event* tag: OCCURRENCE, PERCEPTION, REPORTING, STATE, LSTATE, LACTION, and NONE. The NONE represents that the corresponding morpheme does not belong to the extent, and the other *classes* follow the same definition of TimeML. Similar to the *timex3* extractor, we take two approaches: a set of 26 rules and machine-learning models (e.g., CRF and MEM), based on the set of features used in the *timex3* extractor.

There are several verbs 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 eventual meaning. We define a set of such verbal morphemes (e.g., ‘위하[wi-ha]’(for), and ‘통하[tong-ha]’(through)), to avoid generating meaningless *event* tags.

### 3.2.3 Makeinstance extractor

The goal of *makeinstance* extractor is to generate at least one *makeinstance* tag for each *event* tag, and find appropriate attributes. As we observed that there is only one *makeinstance* tag for each *event* tag in most cases, the *makeinstance* extractor simply generates one *makeinstance* tag for each *event* tag. For the attribute *POS*, we simply take the POS tags obtained from the Korean analyzer. We define a set of 5 rules for the attribute *tense*, and 2 rules for the attribute *polarity*.

### 3.2.4 Tlink extractor

The goal of *tlink* extractor is to make a linkage between two tags, and find appropriate types of the links. For each pair of tags, it determines whether there must be a linkage between them, and finds the most appropriate *relType*. There are 11 *relTypes*: BEFORE, AFTER, INCLUDES, DUR-

Table 3: Features in the two kinds of classifiers.

Features for TM <i>tlink</i>
Surrounding morphemes of the argument tags
Linear order of the argument tags
Attribute <i>type</i> of <i>timex3</i> tag
Attribute <i>class</i> of <i>event</i> tag
Attribute <i>polarity</i> of <i>makeinstance</i> tag
Attribute <i>tense</i> of <i>makeinstance</i> tag
Whether <i>tlink</i> tag is non-consuming tag or not
Does <i>timex3</i> tag exist between argument tags?
Does <i>event</i> tag exist between argument tags?
Is <i>event</i> tag an objective of other <i>event</i> tag in dependency tree?
Features for MM <i>tlink</i>
Surrounding morphemes of <i>event</i> tags
Linear order of <i>event</i> tags
Attribute <i>class</i> of <i>event</i> tags
Attribute <i>polarity</i> of <i>makeinstance</i> tags
Attribute <i>tense</i> of <i>makeinstance</i> tags
Does <i>timex3</i> tag exist between <i>event</i> tags?
Does <i>event</i> tag exist between <i>event</i> tags?

ING, DURING\_INV, SIMULTANEOUS, IDENTITY, BEGINS, ENDS, OVERLAP, and NONE. The NONE represents that there is no linkage between the two argument tags, and the OVERLAP means that the temporal intervals of two tags are overlapping. The other *relTypes* follow the same definition of TimeML. Thus, it is essentially a classification over 11 classes for each pair of two argument tags.

The *tlink* extractor generates intra-sentence *tlinks* and inter-sentence *tlinks*. For the intra-sentence *tlinks*, we take two approaches: a set of 19 rules and machine-learning models (e.g., SVM and LR). Among the four kinds of *tlinks* (e.g., TT *tlink*, TM *tlink*, MM *tlink*, and DM *tlink*), our *tlink* extractor generates the first three kinds of *tlinks*. The reason for excluding DM *tlink* is that we maintain the temporal context initialized with DCT in the *timex3* extractor, so it is not necessary to generate DM *tlinks*. The TT *tlinks* are extracted by comparing normalized *values* of two *timex3* tags. Two models are independently trained for predicting TM *tlinks* and MM *tlinks*, respectively. We tried many possible combinations of features to reach better performance, and obtained sets of features as described in Table 3.

Given a pair of two *makeinstance* tags, it is straight forward to derive *relType* when the two

*event* tags are linked with *timex3* tags. Thus, firstly we predict TT *tlinks*, and thereafter predict TM *tlinks* and MM *tlinks*. For the inter-sentence *tlinks*, we generate MM *tlinks* between adjacent sentences 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).

### 3.3 Online LIFE

As the performance of the Korean analyzer is not stable, we need complementary features to make better classifiers. Jeong and Choi (2015) proposed Language Independent Feature Extractor (LIFE) which generates a pair of class label and topic label for each Letter-Sequence (LS), where LS represents frequently appeared letter sequence. The class labels can be used as syntactic features, while the topic labels can be used as semantic features. The concept of LS makes it language independent, so it is basically applicable to any language. This is especially helpful to some languages that have no stable feature extractors. Korean is one of such languages, so we employ the LIFE to generate complementary features.

The temporal information extractor must work online because it usually takes a stream of documents as an input. However, as the LIFE is originally designed to work offline, we propose an extended version of the LIFE, namely, Online LIFE (O-LIFE), whose parameters are estimated incrementally. When we design O-LIFE, the LS concept of LIFE becomes a problem because the LS dictionary changes. For example, if the LS dictionary has only one LS *goes* and a new token *go* comes in, then the LS dictionary may contain *go* and *es*. Note that the LS *goes* does not exist in the new dictionary. This issue is addressed by our proposed algorithm which basically distributes the values of previously estimated parameters to new prior parameters of overlapping LSs. For the above example,  $\phi_{k, 'goes'}$  will be distributed to  $\beta_{k, 'go'}$  and  $\beta_{k, 'es'}$ , where  $k$  is a topic index.

The formal algorithm of O-LIFE is shown in Algorithm 1, where  $C$  is the number of classes and  $T$  is the number of topics.  $S_{stream}$  is the number of streams, and  $S^s$  is  $s$ -th stream of  $D^s$  documents. The four parameters  $a, b_t, g$  and  $b_c$  are default values of the priors  $\alpha, \beta, \gamma$  and  $\delta$ . The three threshold parameters  $t_1, t_2$ , and  $t_3$  are used to generate LS dictionary.

---

#### Algorithm 1 Online LIFE

---

```

1: INPUT:  $a; b_t; g; b_c; w^p; S^s; t_1; t_2; t_3$ 
2: for  $s=1$  to  $S_{stream}$  do
3:    $\text{Dic}^s = \text{DictionaryGenerator}(t_1, t_2, t_3)$ 
4:   if  $s=1$  then
5:      $\beta_t^s = b_t, 1 \leq t \leq T$ 
6:      $\delta_c^s = b_c, 1 \leq c \leq C$ 
7:   else
8:      $\beta_{t,w}^s = b_t, w \in \text{Dic}^s$ 
9:      $\delta_{c,w}^s = b_c, w \in \text{Dic}^s$ 
10:    for each item  $w_i \in \text{Dic}^{s-1}$  do
11:      if  $w_i \in \text{Dic}^s$  then
12:         $\beta_{t,w_i}^s += \mathbf{B}_{t,w_i}^{s-1} w^p, 1 \leq t \leq T$ 
13:         $\delta_{c,w_i}^s += \mathbf{D}_{c,w_i}^{s-1} w^p, 1 \leq c \leq C$ 
14:      end if
15:      for each  $w'_i \in \text{Dic}^s$  do
16:         $\beta_{t,w'_i}^s += \mathbf{B}_{t,w'_i}^{s-1} w^p, 1 \leq t \leq T$ 
17:         $\delta_{c,w'_i}^s += \mathbf{D}_{c,w'_i}^{s-1} w^p, 1 \leq c \leq C$ 
18:      end for
19:      for each  $w''_i \in \text{Dic}^s$  do
20:         $r = |w_{overlap}| / |w_i|$ 
21:         $\beta_{t,w''_i}^s += \mathbf{B}_{t,w''_i}^{s-1} r w^p, 1 \leq t \leq T$ 
22:         $\delta_{c,w''_i}^s += \mathbf{D}_{c,w''_i}^{s-1} r w^p, 1 \leq c \leq C$ 
23:      end for
24:    end for
25:  end if
26:   $\alpha_d^s = a, 1 \leq d \leq D^s$ 
27:   $\gamma_c^s = g, 1 \leq c \leq C$ 
28:  initialize  $\phi^s, \eta^s, \pi^s$ , and  $\theta^s$  to zeros
29:  initialize class/topic assignments
30:   $[\phi^s, \eta^s, \pi^s, \theta^s] =$ 
31:    ParameterEstimation( $S^s, \alpha^s, \beta^s, \gamma^s, \delta^s$ )
32:   $\mathbf{B}_t^s = \mathbf{B}_t^{s-1} \cup \phi_t^s, 1 \leq t \leq T$ 
33:   $\mathbf{D}_c^s = \mathbf{D}_c^{s-1} \cup \eta_c^s, 1 \leq c \leq C$ 
34: end for

```

---

The LS dictionary of O-LIFE is updated as it reads data. That is,  $|\text{Dic}^{\text{prev}}| \leq |\text{Dic}^{\text{cur}}|$  and  $\text{Dic}^{\text{prev}} \not\subseteq \text{Dic}^{\text{cur}}$ , where  $\text{Dic}^{\text{prev}}$  is the previous dictionary and  $\text{Dic}^{\text{cur}}$  is the current dictionary. To handle this change in dictionary, the two parameters (e.g.,  $\beta, \delta$ ) are updated by four steps, based on an assumption that every unique LS  $w_i$  of the previous dictionary should contribute to the new dictionary as much as possible. Firstly, as described in 8th line of the algorithm, the two parameters are initialized with default values. Secondly, if a particular LS  $w_i$  of the previous dictionary exists in the new dictionary, then the two parameters increase by  $\mathbf{B}_{t,w_i}^{s-1} w^p$  and  $\mathbf{D}_{c,w_i}^{s-1} w^p$ , re-

Table 4: The statistics of Korean TimeBank, where the digits represent the number of corresponding items.

	Training	Validation	Test
documents	536	131	173
sentences	2357	466	879
timex3	1245	253	494
event	6594	1145	2609
makeinstance	6615	1155	2613
tlink	1295	374	674

spectively.  $\mathbf{B}_t^{s-1}$  denotes an evolutionary matrix whose columns are LS-topic distribution  $\phi_t^{s-1}$ , and  $\mathbf{D}_c^{s-1}$  means an evolutionary matrix whose columns are LS-class distribution  $\eta_t^{s-1}$ . By multiplying them with the weight vector  $w^p$ , the contribution in initializing priors is determined for each time slice. We call this value, the *weighted contribution*. Thirdly, for every  $w'_i$  which contains  $w_i$ , the two parameters increase by the weighted contribution. Lastly, for every  $w''_i$  which overlaps with  $w_i$ , the two parameters increase by  $r$  times of the weighted contribution, where  $w_{overlap}$  is the overlapping part.

The class labels and topic labels are converted to morpheme-level features by concatenating labels of LSs overlapping with a given morpheme. We call these features as *LIFE features*, and these are used to train machine-learning models, together with the features based on the raw features.

## 4 Experiments

The Korean TimeBank is divided into a training dataset, a validation dataset, and a test dataset. The statistics of the dataset are described in Table 4. As shown in the table, only one *makeinstance* tag exists for each *event* tag in most cases, which follows the assumption of *makeinstance* extractor.

### 4.1 Timex3 prediction

For the extent and *type* prediction, only the exactly predicted extents and *types* are regarded as correct, and the results are summarized in Table 5, where MEM is trained only with the features generated from raw features, and MEM<sub>L</sub> is trained with both of the features and LIFE features. We employ the CRF++ library<sup>2</sup> and MEM toolkit<sup>3</sup>. 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. The parameter setting for O-LIFE is as follows:  $a=0.1$ ,  $b_t=0.1$ ,  $g=0.1$ ,  $b_c=0.1$ ,  $w^p=(1)$ ,  $C=10$ ,  $T=10$ ,  $t_1=0.4$ ,  $t_2=0.4$ ,  $t_3=0.7$ , and the number of iterations for estimation is 1000.

As shown in the table, CRF gives generally better performance than MEM and rules. We tried a combination of the rules and machine-learning models, and the combination led to an increase in the performance. For example, the combination of CRF and rules gives better performance than using only CRF. This can be explained that there are some patterns that the machine-learning models could not capture, so the combination with the rules can deal with the patterns. Note that using the LIFE features dramatically increases the performance. We believe that this is due to the raw features of Korean (e.g., POS tagger) being unstable. The LIFE features complement these unstable features by capturing syntactic/semantic patterns that are inherent in the given documents. Furthermore, we observed that the combination of the rules and machine-learning models trained with LIFE features does not contribute to the performance. This implies that using LIFE features allows the machine-learning models to capture the patterns that could not be captured by the machine-learning models without LIFE features. The CRF<sub>L</sub> is discovered to be the best, and the other remaining attributes are predicted using the rules. The performance is measured in a sequential manner, so the performance generally decreases from the top to bottom of the table.

Another experiments of *timex3* prediction using TempEval-2 Korean dataset are conducted to compare with the existing method of Angeli and Uszkoreit (2013). The existing method makes use of a latent parse conveying a language-flexible representation of time, and extract features over both the parse and associated temporal semantics. The results of comparison are shown in Table 6, where our method uses CRF<sub>L</sub> for *type* and rules for *value*. For a fair comparison, both methods are trained and tested using only TempEval-2 Korean dataset. As mentioned before, TempEval-2 Korean dataset contained errors, so we corrected them and

<sup>2</sup><http://crfpp.googlecode.com/svn/trunk/doc/index.html>

<sup>3</sup>[http://homepages.inf.ed.ac.uk/lzhang10/maxent\\_toolkit](http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit)

Table 5: Timex3 prediction results, where P represents precision, R means recall, F represents F1 score, points represent *beginPoint/endPoint*, and cal represents *calendar*.

Attri- butes	Performances			
	Comb	P	R	F
extent	Rules	74.79	70.95	72.82
	MEM	31.79	25.1	28.05
	CRF	80.34	65.42	72.11
	MEM,Rules	70.61	70.75	70.68
	CRF,Rules	73.05	<b>72.33</b>	72.69
	MEM <sub>L</sub>	33.51	41.34	37.02
	<b>CRF<sub>L</sub></b>	<b>81.15</b>	<b>72.33</b>	<b>76.49</b>
type	Rules	72.71	68.97	70.79
	MEM	30.26	23.89	26.7
	CRF	<b>78.88</b>	64.23	70.81
	MEM,Rules	68.64	68.77	68.71
	CRF,Rules	71.06	<b>70.36</b>	70.7
	MEM <sub>L</sub>	31.88	39.16	35.15
	<b>CRF<sub>L</sub></b>	<b>75.83</b>	<b>67.59</b>	<b>71.47</b>
value	Rules	71.18	63.44	67.08
points	Rules	71.18	63.44	67.08
freq	Rules	71.18	63.44	67.08
mod	Rules	70.07	62.45	66.04
cal	Rules	70.07	62.45	66.04

Table 6: Results of *timex3* prediction with TempEval-2 Korean dataset, where the digits represent accuracy.

Attributes	Angeli’s method	Our method
type	82.0	<b>100.0</b>
value	42.0	<b>100.0</b>

used for this experiment. As shown in the table, our method gives much better accuracy than the existing method.

## 4.2 Event prediction

The results of *event* prediction are summarized in Table 7, which are similar to the results of *timex3* prediction. By a grid search, the optimal parameter settings are found to be the same as that of the *timex3* extractor. Employing the LIFE features again increases the performance, and the CRF<sub>L</sub> is discovered to be the best.

Table 7: Event prediction results.

Attri- butes	Performances			
	Comb	P	R	F
extent	Rules	75.62	44.0	55.63
	MEM	33.22	15.07	20.73
	CRF	45.33	35.72	39.96
	MEM,Rules	43.57	47.15	45.29
	CRF,Rules	72.67	64.32	68.24
	MEM <sub>L</sub>	37.49	56.34	45.02
	<b>CRF<sub>L</sub></b>	<b>86.49</b>	<b>78.5</b>	<b>82.3</b>
class	Rules	63.97	37.22	47.06
	MEM	31.11	14.12	19.41
	CRF	34.63	27.29	30.53
	MEM,Rules	36.91	39.94	38.37
	CRF,Rules	61.15	54.12	57.42
	MEM <sub>L</sub>	34.74	51.46	41.48
	<b>CRF<sub>L</sub></b>	<b>78.63</b>	<b>71.37</b>	<b>74.82</b>

Table 8: Makeinstance prediction results.

Attri- butes	Performances		
	P	R	F
eventID	86.28	78.19	82.03
polarity	93.59	73.17	82.13
tense	71.03	51.97	60.02

## 4.3 Makeinstance prediction

All the attributes of *makeinstance* tag are predicted through hand-crafted rules. The performance is summarized in Table 8. The measurement of the attribute *POS* is excluded because we simply take the results of the Korean analyzer.

## 4.4 Tlink prediction

By performing a grid search with the validation set, we found that Support Vector Machine (SVM) of C-SVC type with Radial Basis Function (RBF) gives the best performance when  $\Gamma$  of kernel function is  $1/\text{number of features}$ . It is also found that the L1-regularized Logistic Regression (LR) with  $C=1$  gives the best performance. We observed that both models give better performance when a window size is 1.

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 9, where the performance is measured in a sequential manner. As shown in the table, we tried a combina-



Table 9: Tlink prediction results given correct *timex3*, *event*, and *makeinstance* tags.

Attributes	Performances			
	Comb	P	R	F
linkage	SVM	<b>63.84</b>	16.72	26.49
	LR	20.02	63.91	30.49
	Rules	39.11	59.76	47.28
	<b>SVM,Rules</b>	39.04	61.69	<b>47.82</b>
	LR,Rules	21.13	<b>77.22</b>	33.19
rel-Type	SVM	<b>63.84</b>	16.72	26.49
	LR	19.18	61.24	29.22
	Rules	37.27	56.95	45.06
	<b>SVM,Rules</b>	37.27	58.88	<b>45.64</b>
	LR,Rules	20.2	<b>73.82</b>	31.72

Table 10: Tlink prediction results of the combination of SVM and rules, given *timex3*, *event*, and *makeinstance* tags which are predicted using LIFE features.

Attributes	Performances		
	P	R	F
linkage	34.39	38.46	36.31
relType	32.94	36.83	34.77

tion of the rules and machine-learning models, and the combination of SVM and rules performed the best. Note that we do not use the LIFE features for *tlink* prediction because we observed that the LIFE features do not contribute to the performance for *tlink* prediction. We believe that this is because the LIFE features represent only the syntactic/semantic patterns of the given terms, but not arbitrary relations between the terms.

The results of the second case are obtained using the best combination, and are shown in Table 10. Note that the *tlink* tags are predicted without the LIFE features, while the other tags (e.g., *timex3*, *event*, *makeinstance*) are obtained using the LIFE features. To measure the impact of the LIFE features, we also conduct the experiment of the second case given the predicted tags without using the LIFE features, and the results are shown in Table 11. As shown in Table 10 and Table 11, using the LIFE features increases the F1 score about 6 percents.

Table 11: Tlink prediction results of the combination of SVM and rules, given the tags predicted without using the LIFE features.

Attributes	Performances		
	P	R	F
linkage	25.41	36.69	30.02
relType	24.18	34.91	28.57

## 5 Conclusion

We introduced a new method for extracting temporal information from unstructured Korean texts. Korean language has a complex grammar, so there were many research issues to address prior to achieving our goal. We presented such issues and our solutions to them. Experimental results illustrated the effectiveness of our method, especially when we adopted the extended probabilistical model, Online LIFE (O-LIFE), to generate complementary features for training machine-learning models. In addition, as there were no sufficient data for this study, we have manually constructed the Korean TimeBank consisting of more than 3,700 annotated sentences. We will extend our study to interact with Knowledge-Base for achieving better prediction of temporal information.

## Acknowledgments

This work was supported by ICT R&D program of MSIP/IITP. [R0101-15-0062, Development of Knowledge Evolutionary WiseQA Platform Technology for Human Knowledge Augmented Services] Many thanks to my family (HoYoun, Youn-Seo), and my parents.

## References

- Gabor Angeli and Jakob Uszkoreit. 2013. Language-independent discriminative parsing of temporal expressions. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria.
- Steven Bethard. 2013a. Cleartk-timeml: A minimalist approach to tempeval 2013. In *Proceedings of the Seventh International Workshop on Semantic Evaluation*, pages 10–14, Atlanta, Georgia, USA.
- Steven Bethard. 2013b. A synchronous context free grammar for time normalization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 821–826, Seattle, USA.
- Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth Annual Workshop on Computational Learning Theory*, pages 144–152, Pittsburgh, PA, USA.
- Nathanael Chambers, Shan Wang, and Dan Jurafsky. 2007. Classifying temporal relations between events. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 173–176, Prague, Czech Republic.
- Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20(3):273–297.
- Young-Seob Jeong and Ho-Jin Choi. 2015. Language independent feature extractor. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 4170–4171, Texas, USA.
- John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, Williamstown, USA.
- Soojong Lim, ChangKi Lee, Jeong Hur, and Myoung-Gil Jang. 2006. Syntax analysis of enumeration type and parallel type using maximum entropy model. In *Proceedings of the Korea Human-Computer Interaction Conference*, pages 1240–1245.
- Xiao Ling and Daniel S. Weld. 2010. Temporal information extraction. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, Atlanta, Georgia, USA.
- Hector Llorens, Estela Saquete, and Borja Navarro. 2010. Tipsem (english and spanish): Evaluating crfs and semantic roles in tempeval-2. In *Proceedings of the Fifth International Workshop on Semantic Evaluation*, pages 284–291, Uppsala, Sweden.
- Seyed Abolghasem Mirroshandel and Gholamreza Ghassem-Sani. 2012. Towards unsupervised learning of temporal relations between events. *Journal of Artificial Intelligence Research*, 45(1):125–163.
- James Pustejovsky, Jose Castano, Robert Ingria, Roser Sauri, Robert Gaizauskas, Andrea Setzer, and Graham Katz. 2003. Timeml: Robust specification of event and temporal expressions in text. In *New Directions in Question Answering*, pages 28–34, Stanford, USA.
- Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. 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*, pages 127–132, Lisbon, Portugal.
- Jannik Strötgen and Michael Gertz. 2010. Heildetime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the Fifth International Workshop on Semantic Evaluation*, pages 321–324, Uppsala, Sweden.
- Naushad UzZaman and James Allen. 2010. Event and temporal expression extraction from raw text: First step towards a temporally aware system. *International Journal of Semantic Computing*, 4(4):487–508.
- Naushad UzZaman, Hector Llorens, Leon Derczynski, Marc Verhagen, James Allen, and James Pustejovsky. 2013. Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Proceedings of the Seventh International Workshop on Semantic Evaluation*, pages 1–9, Atlanta, Georgia, USA.
- Marc Verhagen, Robert J. Gaizauskas, Frank Schilder, Mark Hepple, Jessica Moszkowicz, and James Pustejovsky. 2009. The tempeval challenge: Identifying temporal relations in text. *Language Resources and Evaluation*, 43(2):161–179.
- Marc Verhagen, Roser Sauri, Tommaso Caselli, and James Pustejovsky. 2010. Semeval-2010 task 13: Tempeval-2. In *Proceedings of the Fifth International Workshop on Semantic Evaluation*, pages 57–62, Uppsala, Sweden.
- Katsumasa Yoshikawa, Sebastian Riedel, Masayuki Asahara, and Yuji Matsumoto. 2009. Jointly identifying temporal relations with markov logic. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 405–413, Suntec, Singapore.