Computers, Materials & Continua
DOI:10.32604/cmc.2021.015636
Article

Tech Science Press

Time-Aware PolarisX: Auto-Growing Knowledge Graph

Yeon-Sun Ahn and Ok-Ran Jeong*

Department of Software, Gachon University, Gyeonggi-do, 13120, Korea
*Corresponding Author: Ok-Ran Jeong. Email: orjeong@gachon.ac.kr
Received: 15 November 2020; Accepted: 19 December 2020

Abstract: A knowledge graph is a structured graph in which data obtained from multiple sources are standardized to acquire and integrate human knowledge. Research is being actively conducted to cover a wide variety of knowledge, as it can be applied to applications that help humans. However, existing researches are constructing knowledge graphs without the time information that knowledge implies. Knowledge stored without time information becomes outdated over time, and in the future, the possibility of knowledge being false or meaningful changes is excluded. As a result, they can't reflect information that changes dynamically, and they can't accept information that has newly emerged. To solve this problem, this paper proposes Time-Aware PolarisX, an automatically extended knowledge graph including time information. Time-Aware PolarisX constructed a BERT model with a relation extractor and an ensemble NER model including a time tag with an entity extractor to extract knowledge consisting of subject, relation, and object from unstructured text. Through two application experiments, it shows that the proposed system overcomes the limitations of existing systems that do not consider time information when applied to an application such as a chatbot. Also, we verify that the accuracy of the extraction model is improved through a comparative experiment with the existing model.

Keywords: Machine learning; natural language processing; knowledge graph; time-aware; information extraction

1 Introduction

Humans acquire knowledge through linguistic processing in which information is learned, thought, and answered questions [1]. Various studies have been conducted for a long time that allow machines to acquire knowledge by imitating these human language processes. One of them, the knowledge graph, considers knowledge to be the relationship between entities, and expresses knowledge in the form of ⟨h, r, t⟩: the triple of 'head entity,' 'relation,' 'tail entity.' In the knowledge graph, each entity is the node and the relation is an edge. This series of information can be expressed in a graph structure and further derived new information. For this reason, knowledge graph has become the core of NLP tasks such as Q&A and semantic analysis.

However, there are two limitations to the existing knowledge graph. First of all, it is static [2]. The popular knowledge graphs such as ConcepNet [3], YAGO3 [4], and NELL [5] are constructed using wide-scale data from Wikidata [6] or various web sources. Although they have hundreds to thousands of entities and relations, they have a limitation that they cannot cover all human knowledge because stored knowledge is fixed. Second, when knowledge graphs are used in actual NLP tasks such as information extraction, Q&A, chatbot system, the existing knowledge graphs derive output without considering time information as a result [7]. For example, 'Galaxy' as a smartphone of 'Samsung' is often referred to rather than the 'universe' or movie title 'Guardians of the Galaxy' in recent, so Samsung smartphones will be derived as the meaning of 'Galaxy' stored in the knowledge graph. At this point, when querying a knowledge graph with 'New Galaxy Series Camera,' if it does not take into account time information, since it is not known which of the various Galaxy series is the 'New Galaxy series,' it answers all Galaxy series camera information or gives nothing at all. In other words, existing knowledge graphs do not reflect information that changes over time and newly emerging information.

In this paper, we propose Time-Aware PolarisX, a system that automatically extends knowledge graphs, including time information. Time-Aware PolarisX is based on a PolarisX [8] system that extracts information from the web in real-time to add newly created or changed information over time in the knowledge graph. To include time information in knowledge extraction from the web, a new Ensemble NER model was established to extract entities and time indicator, and the relation extraction model in PolarisX was relearned using a time-aware knowledge graph as training data. Comparison experiments confirmed that the performance of our Ensemble NER model of Time-Aware PolarisX was improved compared to that of the existing NER model and demonstrated the expandability of the temporal knowledge graph with an example of an entity that changes over time.

2 Related Work

2.1 Knowledge Graphs

A knowledge graph is a structured form of human knowledge, representing the facts composed of entities, relations, and semantic descriptions in a graph. Knowledge graph-based applications such as recommendation systems, QA are effective in solving human problems because they can understand and infer natural language. Among these knowledge graphs, five well-known large-scale knowledge graphs are introduced. Wikidata, DBpedia [9], and YAGO3 are all open source knowledge graphs built from web data including Wikipedia [10], of which YAGO3 is characterized by hierarchical classification of entities. Another knowledge graph, NELL, is a semi-supervised learning system that continues to expand through text pattern learning that extracts knowledge from various web sources. ConceptNet is a graph of human general commonsense built from a variety of sources, including expert-created resources, cropped-source, etc.

Of the five knowledge graphs mentioned earlier, four knowledge graphs, excluding YAGO3, do not deal with the fact that data are not only outdated but also differ in authenticity over time. Studies are being conducted to improve the ambiguity of data by adding temporal information to knowledge. Most of these studies use embedding to model the meaning of knowledge graphs. Knowledge graph completion, link prediction are methods of inferring new relationships that are not in the knowledge graph and adding them. There are ways to learn the connectivity between entities, relation, and time from knowledge graphs that contain some time information, such as YAGO3, as a translation-based scoring function, or, if there is no time information, establish a relational embedding model to learn the sequence in which the relationships occurred. Leblay [8]

proposed relational embedding models incorporating time information to express temporal knowledge graphs in vector space to learn temporal meta-data from incomplete knowledge graphs and applied a factorization machine to improve the scalability and performance. Jiang et al. [11] defined the sequence between constraints-considered relations by pointing out the limitations of any dataset with less time information and defining them as knowledge graph completion problem. For example, learn that a 'wasBornIn' relationship for the same entity is a thing that happened earlier than a 'wasSpouseOf' relationship. Talukdar et al. [12] proposes GraphOrder, a graph-based label propagation algorithm, to learn the sequence between the relationships, as in the paper above. Esteban et al. [13] predicts new relationships from existing relationships by establishing a separate event graph for facts that are likely to be false over time. Garcia-Duran et al. [14] uses temporal tokens (e.g., 'until,' 'since' and '2010') to encode the type of relationship.

PolarisX is one of the Polaris project aimed at automatically expanding knowledge graphs, big data analysis and prediction, and, moreover, building a framework for interactive AI. These PolarisX show higher coverage for neologism, human common sense compared to ConceptNet, a kind of semantic network. Because existing knowledge graphs use a certain amount of data to construct a graph, they cannot deal with new words that have been created or have new meanings after the graph is constructed. On the other hand, PolarisX has the advantage of being able to deal with new words because it extracts knowledge from social media data and news data in real-time and expands the knowledge graph. For this reason, Time-Aware PolarisX used PolarisX as the basic framework and, likewise, belongs to Polaris project [15].

Existing knowledge graphs have limitations that all knowledge is considered as 'fact' because they can only infer relationships between entities and define the order of relationships, but do not know exactly when the event occurred. These limitations are evident in Q&A applications. Without considering the temporal validity of the query, the answer is different from the information human expects to obtain. Furthermore, events and relationships that were created after the knowledge graph was constructed are not detectable. To address these limitations, this paper proposes Time-Aware PolarisX, which extracts new entities and relationships in real-time through external resources and expands knowledge graphs.

2.2 Named Entity Recognition

Named Entity Recognition (NER) is a sub-task of Information Extraction (IE) task, one of the Natural Language Processing(NLP) tasks, which categorizes the named entities of a sentence into defined categories [16]. For example, it may be classified into some categories such as people, organization, location, time, money, etc. There are some open-source APIs for NER such as NLTK [17], StanfordNLP [18]. Or some challenging tasks are being carried out with datasets such as CoNLL2003 [19], W-NUT 17' [20] constructed from various web sources. The state-of-the-art models in the field of NER are mainly models using pre-trained models such as BERT or self-attention mechanism such as LUCK [21], ACE [22], and the type and number of named entity tags are different for each dataset. For example, for CoNLL2003, there are four categories: PER (person), LOC (location), ORG (organization), and MISC (miscellaneous), and for W-NUT 17', there are six categories: Corporation, Creative-work, Group, Location, Person, and Product. In this paper, the NER task was conducted with an ensemble NER [23] model trained in three data sets: CoNLL2003, W-NUT 17 and Groningen Meaning Bank corpus (GMB) [24] to deal with various entities in cross-domain and identify the time indicator words, which are focused in this paper.

2.3 Relation Extraction

Relational extraction (RE) is also a task for extracting structured information from unstructured text, similar to IE, which identifies the semantic relationship between entities and classifies it into previously defined categories. When you receive a sentence and two entities within it as input, RE classifies the relationship between them. Categories can be defined, from equivalence, inclusion, and vertical relationships to more specific relationships such as 'wasMarriedTo' and 'hasWonPrize.' RE is a key component of building knowledge graphs and is used as a major function in NLP applications such as search, sentiment analysis, Q&A, and summarization. In the past, linguistic and lexical features such as POS tagging, syntactic trees, and global decoding constraints were used mainly to extract relationships, but due to the inability of multi-lingual and the disadvantage of a closed-domain environment, currently, pre-trained language models are mainly used [25], and there are many BERT-based models among them (Cohen [26], Zhou [27]). A TACRED [28] dataset consisting of tuples (sentence, entity1, entity2, relation) is typically used in RE. The semantic relationship can be expressed in various formats and languages, in this paper the triple or quadruple format is used. Further details are explained in 3.2.

3 Time-Aware PolarisX

There is a limitation that existing knowledge graphs cannot be expanded after they are built, so the structure is static and does not reflect the time when knowledge occurs. Time-Aware PolarisX builds knowledge graphs automatically, including time information, and continuously expands.

3.1 Motivation

Fig. 1 shows the difference between a knowledge graph that does not consider time and a knowledge graph that considers time. When you ask, 'New Galaxy series' camera' in an application based on knowledge graphs such as chatbot, you can see answers with astronomical telescopes and digital cameras that include 'Galaxy camera' for knowledge graph of the former. On the other hand, for time-considering knowledge graphs of the latter, it allows the expected answer to be given to the user, taking into account the timing of the question.

3.2 Time-Aware PolarisX: Auto-Growing Knowledge Graph

The structure of the entire system is shown in Fig. 2. From a crawling document using News Api in real-time, Keyword Extraction is performed to extract entities with high frequencies of appearance. Use only sentences with extracted keyword entities as input data. This input data is used as input for both entity extraction and relationship extraction. First, in the entity extraction, the ensemble NER model, which can tag time information as well as general named entities, is established and tags them. Next, the BERT-based relation extraction model, semantic analyzer, is used to extract relationships between entities. Use the abbreviation to specify the entities as 'e1, e2,…,' the relation as 'r', and the time as 't'. Depending on the extraction results, triple ⟨e, r, t⟩ is configured for the relationship between entity and time (e.g. ⟨Albert Einstein, wasBornIn, 1879⟩), triple ⟨e1, r, e2⟩ is configured for the relationship between two entities. And if the relationship with two entities and time are extracted, quadruple ⟨e1, r, e2, t⟩ is configured (e.g. ⟨Barak Obama, isMarriedTo, Michelle Robinson, 1992⟩). By considering these created triples or quadruples as a fact and adding them to the graph, the knowledge graph is expanded repeatedly from the process of crawling news data to the construction of knowledge graphs. Ensemble Entity Extractor and Relation Extractor are detailed in 3.2.1 and 3.2.2.

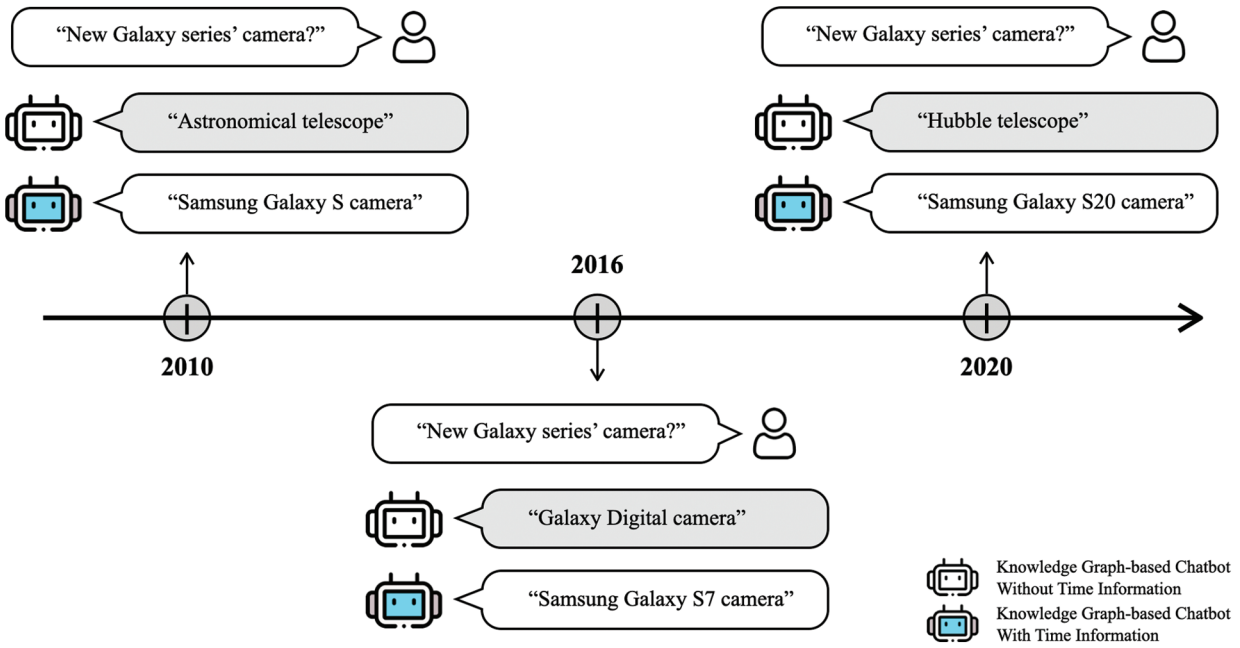

Figure 1: Motivating example

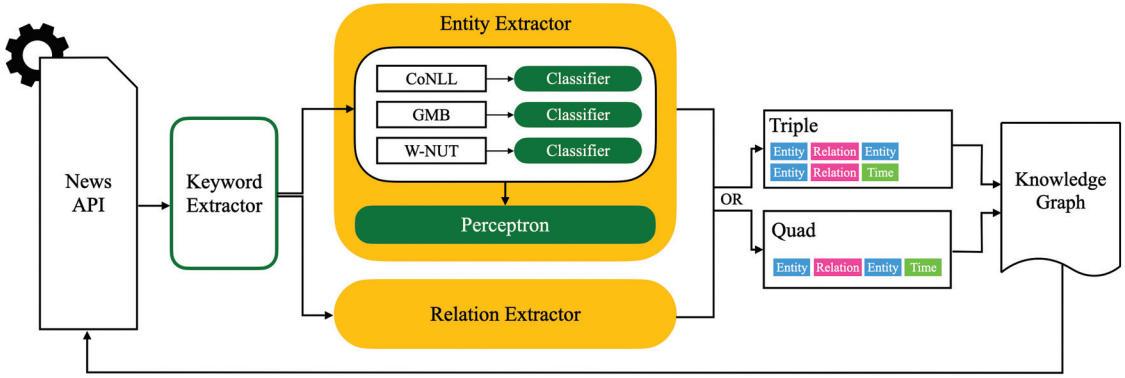

Figure 2: Overall structure

3.2.1 Entity Extractor

Build a classifier to assign a named entity tag to each token from an input sentence. Classifiers enable open-domain learning with datasets created from three different Web sources. Each classifier predicts the probability that each token belongs to specific NER tags, and takes the maximum of them. When three probability values are extracted through three classifiers for each token, the mean of them is used as input into the perceptron layer and it extracts the final NER tag. Eventually, one or more named entities are extracted from each sentence except the 'Others' tag. If only one tag is extracted, it cannot be configured as knowledge, so it is excluded from the results. If there are two tags extracted tags, identify them as a head entity and a tail entity in the order of appearance in the sentence. If two entities and time entity are extracted, the time entity is composed of the last element of the quadruple and the two entities are configured the same as before.

When extracting entities from sentences, a separate time indicator tag is required to extract entities that represent time information. Therefore, the time tag included GMB dataset was used to train the classifier. For a particular token, if the classifier has the highest probability value of belonging to the time tag, the value was input in the perceptron layer instead of the mean value of the three classifiers' so that the time indicator could be extracted accurately. A detailed description of the datasets is given in 4.1.1.

3.2.2 Relation Extractor

In a sentence, there is a main relation that represents the main meaning in context. Semantic Analyzer (or Relations Extractor, RE) modules based on the pre-trained language model BERT were used to extract key relationships between head and tail entities [29]. BERT achieved state-of-the-art in 11 NLP tasks at release and many SOTA models have been constructed based on BERT, so this paper also uses BERT models for RE. An input text is tokenized that BERT can recognize, converted into pre-learned embedding values and input into the model, and specific relation is predicted through the learned model parameters. According to the predicted relation, a triple or quadruple is constructed with the extracted entities. For example, when human and time are extracted with entities, and 'wasBornIn' is extracted as relation, it consists of a triple, when two people and time are extracted with entities, and 'wasMarriedTo' is extracted as relation, it consists of a quadruple.

4 Experiment

Time-Aware PolarisX was built using four datasets and a comparative experiment confirmed that the application was answering appropriately according to the timing of the query, as originally intended in the motivating example. In addition, an experiment using YAGO3 data shows the expandability of the Time-Aware knowledge graph.

4.1 Time-Aware PolarisX Construction

To build a knowledge graph, it is necessary to extract from the data what are the entities and what is the relationship between entities. NER tagging was used for entity extraction and the BERT-based model was used for relation extraction.

4.1.1 Entity Extraction

To build the ensemble NER tagger, the set of data for the Groningen Meeting Bank (GMB) corpus, CoNLL2003, and Noisy User-generated Text (W-NUT 17') was used. The GMB dataset contains a time indicator tag, making it possible to extract time information and other entities together from the text. However, for tags other than time indicator tags, to complement them and create an open-domain environment, two additional datasets were used. CoNLL2003 is a general-purpose dataset widely used in the NER task, and W-NUT 17' dataset is made from Twitter and Youtube data and is used because it contained many new words. To take advantage of all the advantages of each dataset, three classifiers (NER tagger) trained in three datasets each and a multi-layer perceptron [30] were added on top of them to create an ensemble NER model that extracts entities and time information together. Each classifier is based on a pre-trained BERT-based model and each has a fine-tuning, and the perceptron layer is designed to produce one result by processing the results predicted by each of the three classifiers.

Tab. 1 shows the number of words and sentences held by the three datasets GMB, CoNLL2003, W-NUT 17' and data used in the ensemble NER model. Combining the three datasets, the total number of words used in ensemble NER model learning is 269,681, and the

number of sentences used is 15,081. For datasets not used for model training, they were used for model testing.

Table 1: Datasets for NER models

Dataset	Total words	Used words (%)	Total sentences	Used sentences (%)
GMB	1,047,059	261,391 (25%)	47,958	11,990 (25%)
CoNLL2003	36,424	7,163 (20%)	12,735	2,539 (20%)
W-NUT 17'	3,596	1,127 (31%)	1,634	552 (34%)

Ensemble model using multiple datasets requires the process of unifying different named entity tags into a specific standard. In this paper, the tags were unified based on the tags of the commonly used CoNLL data set. However, MISC tags, which mean miscellaneous, were excluded because they could be classified as specific tags in other data sets. As a result, GMB's 'Geo' and 'Gpe' tags were integrated into 'Location,' while the 'Corporation' and 'Group' of the W-NUT 17' data sets were integrated into 'Organization.' The 'Artifact,' 'Natural phenomenon' and 'Event' of the W-NUT 17' data set were classified as 'Others' to unify the tags. The tags in Tab. 2 follow the notations on the dataset sources provided.

Table 2: Named entity tags for each dataset

Dataset	Tags
GMB	Art, Per, Tim, Org, Nat, Eve, Geo, Gpe, O
CoNLL2003	PER, ORG, LOC, MISC, O
W-NUT 17'	Corporation, creative-work, group, location, person, product
For ensemble NER	Person, organization, location, time indicator

NER Tagger's F1-Score, each of which was trained with three data sets, is as follows. The performance of the classifier using CoNLL2003 was 92.8%, that of the classifier using GMB was 70%, and that of the classifier using W-NUT 17' achieved 44.8% accuracy. Tab. 3 shows the accuracy of the ensemble NER tagger by tags combined with these three classifiers, and F1-Score achieved an accuracy of 0.84. 'Others' tag achieved accuracy between 96% and 99% per classifier but was excluded from the accuracy results only to compare the accuracy of the named entity tags with the existing NER models. The dataset for the ensemble NER model showed lower results than the accuracy of the classifier used only the CoNLL2003 dataset because of the addition of GMB and W-NUT 17' data, but 14% more accuracy than those built with GMB and 40% more accuracy than those built with W-NUT 17'.

4.1.2 Relation Extraction

For relationship extraction, the same BERT-based model was used as the PolarisX. However, instead of the TACRED dataset used in PolarisX, Time-Aware PolarisX used the static time-aware KG, YAGO3, for model training. YAGO3 is a semantic knowledge graph built from large-scale WordNet and Wikipedia, with time information included for some data. Time information is expressed as relations representing temporal events such as 'wasBornIn,' 'wasGraduatedFrom,' and

Table 3: Accuracy of proposed ensemble NER model for each tag

Tag	Precision	Recall	F1-Score
Person	0.89	0.90	0.90
Organization	0.77	0.68	0.72
Location	0.83	0.81	0.82
Time indicator	0.94	0.89	0.91
Avg.	**0.86**	**0.82**	**0.84**

'happenedIn.' There are a total of 12,430,701 data instances, of which 4,190,241 data instances, including time information, were used to learn and tune the relation extraction model.

YAGO3 has a total of 37 relationships, but only 21 of them have been used as relation labels, including those representing general facts, those representing temporal events, and those that make them true at a specific point in time. For example, relationships such as 'wasBornIn,' 'hasWonPrize' and 'happenedIn' were used in relation extraction models because they had more exact meanings when they were with time information. However, relationships that represent absolute facts for the passage of time or that are not eventful were excluded from the relationships that are considered, such as 'hasOfficialLanguage' and 'hasCurrency.' Tab. 4 shows the 21 relationships used in this paper among the total 37 relationships of YAGO3.

Table 4: YAGO3 37 relations (selected 21 relations with bold text)

Relations
isLeaderOf, hasOfficialLanguage, imports, dealsWith, hasNeighbor, **isInterestedIn**, exports, hasCurrency, **hasCapital**, hasAcademicAdvisor, **isKnownFor**, **owns**, **isCitizenOf**, **isLocatedIn**, hasMusicalRole, edited, isConnectedTo, **actedIn**, **participatedIn**, isPoliticianOf, wroteMusicFor, hasChild, isAffiliatedTo, hasGender, **playsFor**, **directed**, **influences**, **hasWonPrize**, hasWebsite, **livesIn**, **wasBornIn**, **created**, **diedIn**, **isMarriedTo**, **happenedIn**, **worksAt**, **graduatedFrom**

The YAGO3 dataset contains a lot of information, such as the triple that constitutes the knowledge graph, the entity types in the triple, the relation types, the whole category scheme, and the source link from which the triple was extracted but does not have the original text from which the triple was extracted. To learn the relation extraction model by building datasets with original text with time information, the tag ⟨extractionSource⟩ within the dataset was used. For the purpose of extracting the original text corresponding to the entity of the triple, which contains only time information, the relevant sentences were extracted from Wikipedia page link (e.g. https://en.wikipedia.org/wiki/Galaxy). RE training dataset was constructed with extracted 65,247 sentences from 40,097 triples and used to train the relation extraction model. As mentioned earlier, the relation extraction module of PolarisX was used, but some parameters were tuned according to the created dataset.

4.2 Comparison Experiments

In 4.2.1, comparative experiments with existing knowledge graphs were conducted to demonstrate that the motivating example was achieved as shown in 3.1. 4.2.2 demonstrated performance improvements by comparing the existing SOTA NER models for each dataset and the three NER models constructed to create the proposed ensemble NER model.

4.2.1 Comparison of Search Results for Knowledge Graphs

Fig. 3 compares the results of searching for 'A new Galaxy series' on Wikidata, an existing knowledge graph that does not take time into account, with the results of searching on Time-Aware PolarisX. In the former case, out of a total of 103 search results, Samsung's digital camera model, 'Galaxy Camera,' was 8, and Universe observation-related telescopes were 95 indicating that none of the results related to the Samsung smartphone series were detected. Even the eight 'Galaxy Camera' results, despite having 'New' in the query, show all the results that include 'Galaxy,' indicating that they did not understand the query. Regardless of the timing of the search, data related to 'Galaxy Camera' from 2012 to 2014 were retrieved. Furthermore, in the case of ConceptNet, another open-source knowledge graph, no information about Samsung's smartphone Galaxy was found in the 'Galaxy' search results.

On the other hand, it can be seen that the Time-Aware PolarisX proposed works as expected in the Motivating Example referred to in 3.1. Time-Aware PolarisX expands knowledge graphs with time information, so when searched for the same question in 2020, it showed the most recently mentioned Samsung Galaxy S21 smartphone yet to be released as a result, and it can also be seen that the results for the S20 released this year are derived. As such, Time-Aware PolarisX takes into account time information, allowing users to provide useful knowledge when searching for specific entities that they intend.

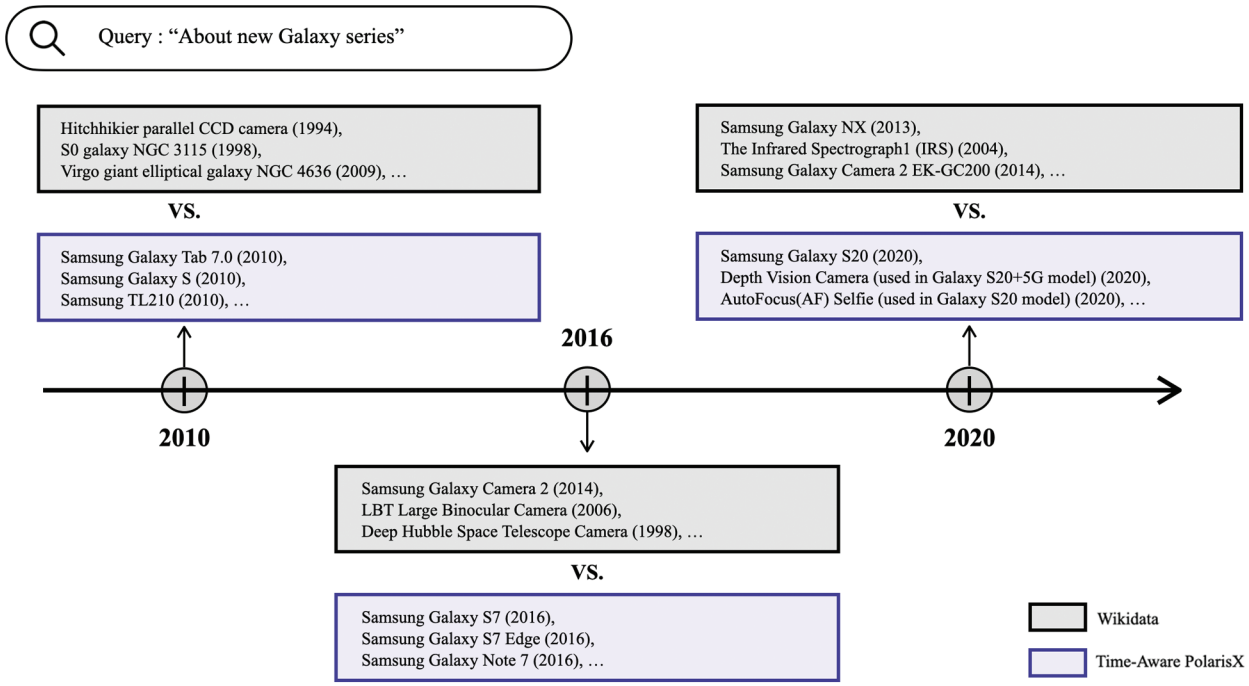

Figure 3: Comparing search results for existing knowledge graph with Time-Aware PolarisX-based knowledge graph

4.2.2 Comparison of NER Models

Tab. 5 compares the performance of each of the three classifiers used in the ensemble NER model described in 4.1.1 to the existing SOTA NER models. As noted in Section 2.2, studies

using self-attention or pre-trained embedding models, such as LUKE, CNN, and BERT, achieve SOTA in the NER task. ACE made various binding forms of pre-trained embedding so that they could be predicted, and Biaffine-NER used a concept of dependency parser. In the case of W-NUT 17', because it includes noise and new words, models using Part-of-Speech Tagging and gazetteer information together achieved SOTA.

Table 5: Comparing accuracy of our model with the existing models for each dataset

GMB	F1-score	CoNLL2003	F1-score	W-NUT 17'	F1-score
CRF [31]	0.46	LUKE	0.94	Arcada [34]	0.399
Bi-LSTM [32]	0.48	ACE	0.936	SJTU-Adapt [35]	0.404
Multi-layer Perceptron	0.60	CNN Large [33]	0.935	SpinningBytes [36]	0.407
BERT	0.67	Biaffine-NER	0.935	UH-RiTUAL [37]	0.418
Our model*	**0.70**	**BERT-Large***	**0.928**	**Our model***	**0.448**

All our three classifiers were modulated to fit the dataset with BERT-based models. For classifiers learned with the CoNLL2003 dataset, it can be seen that F1-Score is slightly lower than the four existing models but not significantly different at around 0.012. In addition, for the GMB and W-NUT 17' datasets, the highest accuracy was achieved compared to the previous models.

4.3 Covering New Knowledge

The following experiments were conducted to ensure that new knowledge was added to the knowledge graph over time. Time-Aware PolarisX, which used YAGO3 as a knowledge graph before expansion, has expanded its knowledge graph to 100 news articles from 2020.11.06 to 2020.11.09 with 2,204 sentences. Fig. 4 shows the relationship changes for 'Joe Biden' before and after the knowledge graph expansion. Of the 2204 sentences, it can be seen that the 'isLeaderOf' relationship has been added, despite the 147 election-related sentences accounting for 6% of the total data. 'Joe Biden' was elected the next U.S. president on November 8, 2020, and an article was reported about it, especially in the sentences such as 'The results are Finally in and the world is reacting to the story of Joe Bidden over the Donald Trump in the U.S.'

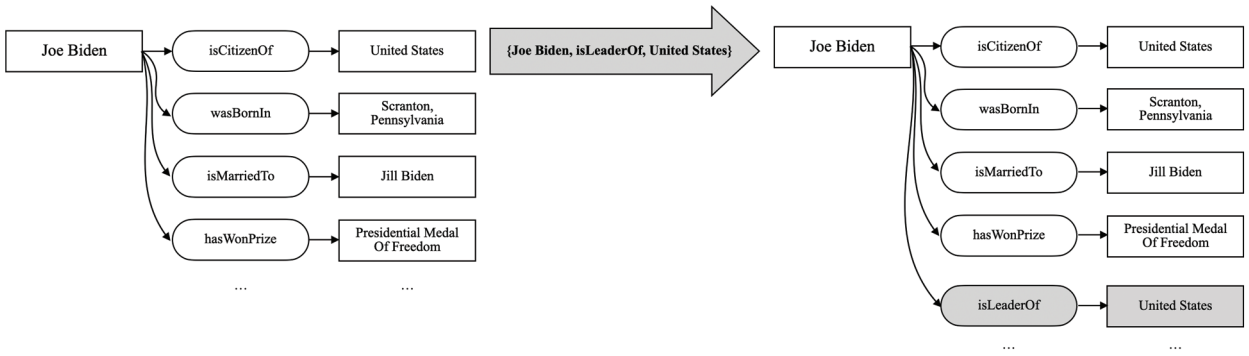

Figure 4: Knowledge graph expansion example

4.4 The Change of Authenticity over Time

This experiment was conducted to verify that Time-Aware Knowledge Graph was able to respond well to the fact that its authenticity varies over time. To see the changes over time, the Time-Aware knowledge graph YAGO3 was set as a baseline and Time-Aware PolarisX expanded with the latest data not in YAGO3 to ensure that it responds well to changes. Tab. 6 lists the entities for Samsung's smartphone Galaxy in YAGO3 and Time-Aware PolarisX according to the year. For YAGO3, data prior to 2009 also exist for 'Galaxy,' meaning other than Samsung Galaxy smartphones, but only the Samsung Galaxy series-related entities are listed in Tab. 6 to ensure that time information and identity are extracted correctly.

Table 6: Comparing YAGO3 with time-aware PolarisX-extended YAGO3

Knowledge graph	Year	Examples of entity
YAGO3	2009∼2011	Samsung_Galaxy_Spica (2009), Samsung_Galaxy_S (2010), Samsung_Galaxy_Note (2011), ...
	2012∼2014	Samsung_Galaxy_S3 (2012), Samsung_Galaxy_S4 (2013), Samsung_Galaxy_Note_4 (2014), ...
	2015∼2017.04	Samsung_Galaxy_S6_Edge (2015), Samsung_Galaxy_J5 (2016), Samsung_Galaxy_S8 (2017), ...
Time-Aware PolarisX	2018	Samsung_Galaxy_AR_Emoji (2018), Samsung_Galaxy_S9 (2018), Samsung_Galaxy_Watch (2018), ...
	2019	Samsung_Galaxy_S10 (2019), Samsung_Galaxy_Buds (2019), Samsung_Galaxy_Fold (2019), ...
	2020	Samsung_Galaxy_S20 (2020), Samsung_Galaxy_S20_Plus (2020), Samsung_Galaxy_Z_Flip (2020), ...

Of the 4,190,241 data that contain time information from 1654 to April 2017, Time-Aware KG, the data that includes 'Galaxy' totaled 414 with 0.009%. Of the 414 data, 157 are meant for Samsung smartphones, accounting for 38%. Time-Aware PolarisX expanded using data from 2018 to 2020 that are not in YAGO3. The data were extracted based on popularity using News API. The extracted data has a total of 9,400 articles and 234,577 sentences. Of the total sentences, 286 included the word "Galaxy," which was 0.121%, and out of 286, 135 data meant "galaxy," accounting for 47.2%.

When searching for 'Galaxy' on two knowledge graphs, Tab. 6 shows that the Galaxy smartphone series until 2017 is searched for YAGO3 and the Time-Aware PolarisX is searched from 2017 to the present. This shows that Time-Aware PolarisX is good at extracting new information whenever new Samsung Galaxy series information is generated through the extraction of entities according to the launch year for Galaxy smartphone series after 2017. Extracting time information and entities together, as shown in Tab. 6, enables users to understand what the new Galaxy series is, depending on the point at which they are asked, and thus results as intended in the question.

5 Conclusion and Future Perspectives

We proposed Time-Aware PolarisX, a system that automatically builds and continuously expands knowledge graphs including time information in existing knowledge, to address the limitations caused by the absence of time information in existing knowledge graphs. The ensemble

NER model showed improved performance compared to the existing NER model while extracting entities and time information together. Several experiments have also shown the scalability of the knowledge graph, comparing the search results of the knowledge graph built with the existing knowledge graph and the proposed system, showing that more useful answers can be obtained from knowledge with time information. Finally, continuous experimentation of knowledge graph with Samsung Galaxy smartphone series as an example proved that both knowledge generated over time and knowledge changing over time can be well dealt with. Because Time-Aware PolarisX builds/extends knowledge graphs with knowledge with time information, it is expected to be more useful than existing knowledge graph-based systems in interactive AI systems. If Time-Aware PolarisX is created, it is expected that by answering the time-valid or most recent knowledge of the question, results that meet the user's intentions and needs can be obtained.

The process of extracting documents from the web is necessary to continuously extract new knowledge, but in many cases, the time information is not clearly visible in the documents. Subsequently, we will study how to automatically learn the sequence of relationships and define the temporal order between knowledge, even if time information is not shown. In addition, it is difficult to verify whether the extracted knowledge is correct because knowledge is extracted in real-time from the raw text when automatically expanding the knowledge graph through web crawling. By addressing these limitations, we will develop into an automatically extended knowledge graph satisfying the confidence.

Funding Statement: This research was supported by Basic Science Research Program through the NRF(National Research Foundation of Korea), the MSIT(Ministry of Science and ICT), Korea, under the National Program for Excellence in SW supervised by the IITP(Institute for Information & communications Technology Promotion) and the Gachon University research fund of 2019(Nos. NRF2019R1A2C1008412, 2015-0-00932, GCU-2019-0773).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Ji, S. Pan, E. Cambria, P. Marttsssinen and P. S. Yu, "A survey on knowledge graphs: Representation, acquisition and applications. arXiv preprint arXiv: 2002.00388, 2020.

[2] R. Trivedi, H. Dai, Y. Wang and L. Song, "Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. arXiv preprint arXiv: 1705.05742, 2017.

[3] C. Havasi, R. Speer and J. Alonso, "ConceptNet: A lexical resource for common sense knowledge," *Recent Advances in Natural Language Processing V: Selected Papers from RANLP*, vol. 6, pp. 269–280, 2007.

[4] F. M. Suchanek, G. Kasneci and G. Weikum, "Yago: A core of semantic knowledge," in *Proc. of the 16th Int. Conf. on World Wide Web*, Banff, Alberta, Canada, pp. 697–706, 2007.

[5] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, B. Yang *et al.,* " Never-ending learning," *Communications of the ACM*, vol. 61, no. 5, pp. 103–115, 2018.

[6] D. Vrandečić, "Wikidata: A new platform for collaborative data collection," in *Proc. of the 21st Int. Conf. on World Wide Web*, Lyon, France, pp. 1063–1064, 2012.

[7] J. Leblay and M. W. Chekol, "Deriving validity time in knowledge graph,"in *Companion Proc. of The Web Conf.*, Lyon, France, pp. 1771–1776, 2018.

[8] S. Yoo and O. Jeong, "Automating the expansion of a knowledge graph," *Expert Systems with Applications*, vol. 141, pp. 112965, 2020.

[9] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak *et al.,* "Dbpedia: A Nucleus for a Web of Open Data," *The Semantic Web*, vol. 4825. Berlin, Heidelberg: Springer, pp. 722735, 2007.

[10] D. Karpf and Wikipedia Blogs, *Second Life, and Beyond: From Production to Produsage*. New York: Axel Bruns, 2009.

[11] T. Jiang, T. Liu, T. Ge, L. Sha, B. Chang *et al.,* "Towards time-aware knowledge graph completion," in *Proc. of COLING 2016, the 26th Int. Conf. on Computational Linguistics: Technical Papers*, Osaka, Japan, pp. 1715–1724, 2016.

[12] P. P. Talukdar, D. Wijaya and T. Mitchell, "Acquiring temporal constraints between relations," in *Proc. of the 21st ACM Int. Conf. on Information and Knowledge Management*, Maui, Hawaii, USA, pp. 992–1001, 2012.

[13] C. Esteban, V. Tresp, Y. Yang, S. Baier and D. Krompaß, "Predicting the co-evolution of event and knowledge graphs," in *2016 19th Int. Conf. on Information Fusion*, Heidelberg, Germany, pp. 98–105, 2016.

[14] A. García-Durán, S. Dumančić and M. Niepert, "Learning sequence encoders for temporal knowledge graph completion," arXiv preprint arXiv: 1809.03202, 2018.

[15] S. Yoo and O. Jeong, "Social media contents based sentiment analysis and prediction system," *Expert Systems with Applications*, vol. 105, pp. 102–111, 2018.

[16] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes*, vol. 30, no. 1, pp. 3–26, 2007.

[17] E. Loper and S. Bird, "NLTK: The natural language toolkit. arXiv preprint cs/0205028, 2002.

[18] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard *et al.,* "The Stanford CoreNLP natural language processing toolkit," in *Proc. of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55–60, 2014.

[19] E. F. Sang and F. De Meulder, "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition," arXiv preprint cs/0306050, 2003.

[20] L. Derczynski, E. Nichols, M. van Erp and N. Limsopatham, "Results of the WNUT2017 shared task on novel and emerging entity recognition," in *Proc. of the 3rd Workshop on Noisy User-generated Text*, Copenhagen, Denmark, pp. 140–147, 2017.

[21] I. Yamada, A. Asai, H. Shindo, H. Takeda, Y. Matsumoto *et al.,* "Deep contextualized entity representations with entity-aware self-attention," arXiv preprint arXiv: 2010.01057, 2020.

[22] X. Wang, Y. Jiang, N. Bach, T. Wang, Z. Huang *et al.,* "Huang etal, Automated concatenation of embeddings for structured prediction. arXiv preprint arXiv: 2010.05006, 2020.

[23] T. G. Dietterich, *Ensemble methods in machine learning, in International workshop on multiple classifier systems*. Berlin, Heidelberg: Springer, pp. 1–15, 2000.

[24] J. Bos, V. Basile, K. Evang, N. J. Venhuizen and J. Bjerva, "The groningen meaning bank," in *Handbook of linguistic annotation*, Dordrecht: Springer, pp. 463–496, 2017.

[25] J. Yu, B. Bohnet and M. Poesio, "Named entity recognition as dependency parsing," arXiv preprint arXiv: 2005.07150, 2020..

[26] A. D. Cohen, S. Rosenman and Y. Goldberg, "Relation extraction as two-way span-prediction," arXiv preprint arXiv: 2010.04829, 2020.

[27] W. Zhou, K. Huang, T. Ma and J. Huang, "Document-level relation extraction with adaptive thresholding and localized context pooling," arXiv preprint arXiv: 2010.11304, 2020.

[28] Y. Zhang, V. Zhong, D. Chen, G. Angeli and C. D. Manning, "Position-aware attention and supervised data improve slot filling," in *Proc. of the 2017 Conf. on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, pp. 35–45, 2007.

[29] J. Devlin, M. W. Chang, K. Lee and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv: 1810.04805, 2018.

[30] S. K. Pal and S. Mitra, "Multilayer perceptron, fuzzy sets classifiaction," *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 683–697, 1992.

[31] C. Sutton and A. McCallum, "An introduction to conditional random fields," *Foundations and Trends in Machine Learning in ACM*, vol. 4, no. 4, pp. 267–373, 2012.

[32] Z. Huang, W. Xu and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," arXiv preprint arXiv: 1508.01991, 2015.

[33] A. Baevski, S. Edunov, Y. Liu, L. Zettlemoyer and M. Auli, "Cloze-driven pretraining of self-attention networks. arXiv preprint arXiv: 1903. 07785, 2019.

[34] P. Jansson and S. Liu, "Distributed representation, LDA topic modelling and deep learning for emerging named entity recognition from social media," in *Proc. of the 3rd Workshop on Noisy User-generated Text*, Copenhagen, Denmark, pp. 154–159, 2017.

[35] B. Y. Lin, F. F. Xu, Z. Luo and K. Zhu, "Multi-channel BiLSTM-CRF model for emerging named entity recognition in social media," in *Proc. of the 3rd Workshop on Noisy User-generated Text*, Copenhagen, Denmark, pp. 160–165, 2017.

[36] P. Von Däniken and M. Cieliebak, "Transfer learning and sentence level features for named entity recognition on tweets," in *3rd Workshop on Noisy User-generated Text (W-NUT), Association for Computational Linguistics*. Copenhagen, Denmark, pp. 166–171, 2017.

[37] G. Aguilar, S. Maharjan, A. P. López-Monroy and T. Solorio, "A multi-task approach for named entity recognition in social media data. arXiv preprint arXiv: 1906.04135, 2019.