IEICE TRANS. INF. & SYST., VOL.E105–D, NO.3 MARCH 2022

PAPER

# Competent Triple Identification for Knowledge Graph Completion under the Open-World Assumption

**Esrat FARJANA**[†a)], **Natthawut KERTKEIDKACHORN**[††], ***Nonmembers***, ***and*** **Ryutaro ICHISE**[†,†††,††††], ***Senior Member***

**SUMMARY** The usefulness and usability of existing knowledge graphs (KGs) are mostly limited because of the incompleteness of knowledge compared to the growing number of facts about the real world. Most existing ontology-based KG completion methods are based on the closed-world assumption, where KGs are fixed. In these methods, entities and relations are defined, and new entity information cannot be easily added. In contrast, in open-world assumptions, entities and relations are not previously defined. Thus there is a vast scope to find new entity information. Despite this, knowledge acquisition under the open-world assumption is challenging because most available knowledge is in a noisy unstructured text format. Nevertheless, Open Information Extraction (OpenIE) systems can extract triples, namely (head text; relation text; tail text), from raw text without any prespecified vocabulary. Such triples contain noisy information that is not essential for KGs. Therefore, to use such triples for the KG completion task, it is necessary to identify competent triples for KGs from the extracted triple set. Here, competent triples are the triples that can contribute to add new information to the existing KGs. In this paper, we propose the Competent Triple Identification (CTID) model for KGs. We also propose two types of feature, namely syntax- and semantic-based features, to identify competent triples from a triple set extracted by a state-of-the-art OpenIE system. We investigate both types of feature and test their effectiveness. It is found that the performance of the proposed features is about 20% better compared to that of the ReVerb system in identifying competent triples.

***key words:*** *knowledge extraction, information retrieval, competent triple, knowledge graph completion*

## 1. Introduction

A knowledge graph (KG) is a multi-relational directed graph representation of a knowledge base (KB). In a KG, we can represent knowledge in the triple format [head entity $h$, relation $r$, tail entity $t$], which expresses an entity-entity relationship. KGs are widely used for various AI-related tasks, such as web search, question-answering, entity linking, and natural language processing. Example of KGs include Wikidata [1], YAGO [2], and Freebase [3]. Although KGs are widely used, with the exponential growth of data, most existing KGs are noisy and incomplete. Available knowledge in KGs is lagging behind available data, which are growing at a rapid pace. Researchers have aimed to improve the accuracy and reliability of KGs by predicting the existence of various relations among entities, which is known as the KG completion task.

An embedding-based model is commonly used in the KG completion task. Existing embedding-based KG completion methods such as TransE [4] and ComplEx [5] are performed under the closed-world assumption, where KGs are fixed, and all entities and relations are already defined. These models, which heavily rely on the structure of existing KGs, can well predict missing relationships between well-connected entities. Because of their high reliance on the structure of existing KGs, it is challenging to add new entity information using similar settings.

In contrast, in the open-world assumption, entities and relations are not defined in advance. Knowledge can thus be added to KGs from natural language text data, which is easily available. About 95% of available data is unstructured text data [6]. It is not possible to extract entity information directly from the natural text because it is unstructured. In this context, the Open Information Extraction (OpenIE) [7]–[9] system extracts a binary relationships in the triple format (e.g., (Barack Obama, was born in, Honolulu)) from unstructured text without any prespecified vocabulary. Although OpenIE does not require any prior knowledge, the quality of OpenIE triples varies. The system is likely to include lots of noisy and redundant information in KBs, making them inconsistent.

We propose a supervised learning model for identifying triples (extracted by the OpenIE system) to add information to existing KGs. For this task, we classify all triples into two classes, namely *competent* and *incompetent* where the former (latter) refers to a triple that is relevant (not relevant) to the context of KG. In this study, we develop syntax- and semantic-based features that facilitate the correct identification of *competent* triples.

The major contributions of our paper are as follows:

- We formulate a new research problem in KG completion area that can measure both the correctness and appropriateness of extracted triples in advance.
- We propose an automated method called Competent Triple Identification (CTID) for identifying *competent* triples to assist the KG completion task by leveraging the OpenIE system. We develop syntax- and semantic-

Manuscript received July 8, 2021.
Manuscript revised September 20, 2021.
Manuscript publicized December 2, 2021.

[†]The authors are with The Graduate University for Advanced Studies, SOKENDAI, Tokyo, 101–8430 Japan.
[††]The author is with Japan Advanced Institute of Science and Technology, Nomi-shi, 923–1292 Japan.
[†††]The author is with National Institute of Informatics, Tokyo, 101–8430 Japan.
[††††]The author is with National Institute of Advanced Industrial Science and Technology, Tokyo, 135–0064 Japan.
a) E-mail: esrat_farjana@nii.ac.jp

DOI: 10.1587/transinf.2021EDP7148

based features to identify *competent* triples.

- We propose a procedure for creating a dataset for the above task with an automated annotation procedure. We also provide the dataset for future research.

The rest of this paper is organized as follows. Section 2 presents the problem definition, and Sect. 3 discusses related works. Section 4 describes the proposed features in detail. Section 5 presents experiments conducted to evaluate the proposed features. Section 6 presents some associated discussion of our experiments, and Sect. 7 concludes the paper.

## 2. Problem Definition

In this study, we consider the extraction of useful knowledge for the KG completion task under the open-world assumption. We define two types of triple, namely *competent* and *incompetent*. The definitions required to define the problem are as follows:

- **Knowledge Graph:** Let $KG = (E, R, \tau)$ be a KG that consists of a large number of facts about the real world, where $E$ denotes the entity set, $R$ denotes the relation set and $\tau$ denotes the triple set. Here, $\tau = (h, r, t)$, where $h$ denotes the head entity, $t$ denotes the tail entity and $r$ denotes the relation between $h$ and $t$.
- **Open-world Assumption:** Let $OWA$ represent the open-world assumption, where all entities and relations do not already exist in KGs. To be more precise, $\exists O_e \notin E$ and $\exists O_r \notin R$ where $O_e$ denotes an open-world entity and $O_r$ denotes an open-world relation. Therefore, $OWA$ contains new entity information that is not present in existing KGs.
- **Competent Triple:** Let $CT = (h, r, t)$ be a competent triple for a given context $c$, where $(h, r, t)$ are related to the context $c$ and $h \notin E$ or $t \notin E$ or, $r \notin R$.
- **Incompetent Triple:** Let $IT = (h, r, t)$ be an incompetent triple, where $(h, r, t)$ are not related to the context $c$.

**Problem (Competent Triple Identification, CTID)** Given (a) a set of reference texts $R_T$, which represents the context $c$ for KG, and (b) a set of sentence texts $S_T$, which represents related knowledge for each context $c$ in an unstructured text format, we use the OpenIE system to extract the triple $t_r$ from each sentence text $s$, where $s \in S_T$. From the extracted triple set $\tau$, we identify *competent* triples, which can be used for KG completion. An illustration of this problem is shown in Fig. 1. Here, we use the OpenIE system to generate triples for each sentence text $s$. We then classify these triples into two classes, namely *competent* and *incompetent*. Here, the first triple, (I; go to; school), is not essential for KG, whereas the second triple, which contains information about the birthplace of Barack Obama, is necessary.

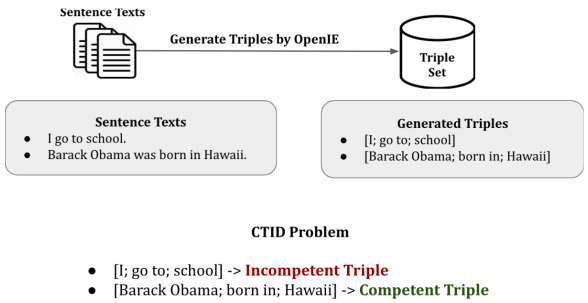



**Fig. 1** Illustration of CTID problem. Triples generated by OpenIE can be noisy. The CTID model can effectively identify competent triples for KGs.

## 3. Related Works

Although our focus is to identify *competent* triples for KG completion, there have been many previous works related to the KG completion task. We can divide those works into two categories. One is the closed-world assumption, where all entities and relations are already known, and another one is the open-world assumption, where all entities and relations are not previously known.

**Closed-world assumption:** Most existing embedding-based models [4], [10]–[12] use the closed-world assumption. These models add missing facts using the existing KB. Link-prediction is used to find a missing relation for existing entities. Other approaches, such as AMIE [13] and GRank [14], are based on rule learning. These approaches use rules to deduce missing facts in a KB. Neither embedding- nor rule-based methods can add new entities or relations for KG completion. For KG refinement, most studies [15], [16] use existing KBs. Therefore, methods based on the closed-world assumption cannot discover facts not contained in a KB.

**Open-world assumption:** Open information extraction systems such as ReVerb [17] and OLLIE [18] extract triples from a sentence based on syntactic and lexical patterns. Although these approaches can extract triples from unstructured text, they cannot measure the importance of the extracted triples to enrich KBs. Additionally, most of the extracted triples contain noisy information. T2KG [19] is an end-to-end system for completing a KG under the open-world assumptions. Although it can populate the KG, it adds incompetent knowledge into the KG.

In addition to the above two categories, some works utilized external resources. Some studies [20], [21] investigated knowledge extraction and entity mapping. The extracted triple is stored as a Resource Description Framework (RDF) triple using WordNet and DBpedia. But it is challenging to add entity information as RDF format from the available raw text data in open-world. Therefore, all elements of the triple are not integrated into a KG. Another approach is ontology-based knowledge extraction [22], where WordNet with a fixed ontology is used. These approaches cannot add knowledge that is not included in the existing

KG. This approach does not identify which triples are essential for KG. To the best of our knowledge, knowledge refinement under the open-world assumption has not been previously studied. Hence, in this study, our main focus is the extraction of competent triples from natural text data that can be used to complete existing KGs.

## 4. Competent Triple Identification

In this study, we propose the CTID model for identifying *competent* triples from a triple set. These triples can assist the completion of existing KGs. Here, we utilize the OpenIE system for extracting triples from unstructured text. We use ReVerb, a state-of-the-art OpenIE system, as our baseline model for the experiments because ReVerb is the base model of other recent OpenIE systems such as OLLIE [18]. In addition, we use features of ReVerb to compare our proposed features because OLLIE utilized the same features. In the next two subsections, we respectively discuss the ReVerb system and the proposed model CTID.

### 4.1 ReVerb System

In our approach, we utilize the syntactic and lexical constraint mechanisms of the ReVerb system [17]. The ReVerb system is designed for web-scale information extraction where relations cannot be prespecified. It automatically identifies triples and extracts binary relationships from English sentences.

The ReVerb system addresses two types of error that occur in OpenIE systems such as TextRunner [23] and WOE [24], namely incoherent extraction and uninformative extraction. For the former, the extracted relation phrase has no meaningful interpretation (e.g., "contains omits", "recalled began"), and for the latter critical information is omitted (e.g., "Faust, made, a deal" for the input sentence "Faust made a deal with the devil").

To avoid incoherent and uninformative extraction, the ReVerb system introduces syntactic and lexical constraints. The syntactic constraint requires the relation phrase to match the part-of-speech (POS) tag pattern shown in Table 1. This pattern states that every multi-word relation phrase must begin with a verb, end with a preposition, and be a contiguous sequence of words in the sentence. The system also introduces a lexical constraint to avoid overspecified relation extraction. The extraction algorithm uses the features shown in Table 2 to assign a confidence score to each extracted triple. The features have weights in the confidence calculation.

**Table 1** ReVerb's POS-based regular expression for reducing incoherent and uninformative extraction

| V \| VP \| VW*P |
|---|
| V = verb particle? adv? |
| W = (noun \| adj \| adv \| pron \| det) |
| P = (prep \| particle \| inf. marker) |

**Table 2** Features used in ReVerb system

| Weight | Feature |
|---|---|
| 1.16 | $(x, r, y)$ covers all words in $s$ |
| 0.50 | The last preposition in $r$ is for |
| 0.49 | The last preposition in $r$ is on |
| 0.46 | The last preposition in $r$ is of |
| 0.43 | $len(s) \leq 10$ words |
| 0.43 | There is a WH-word to the left of $r$ |
| 0.42 | $r$ matches VW*P from Table 1 |
| 0.39 | The last preposition in $r$ is to |
| 0.25 | The last preposition in $r$ is in |
| 0.23 | $10 \text{ words} < len(s) \leq 20 \text{ words}$ |
| 0.21 | $s$ begins with $x$ |
| 0.16 | $y$ is a proper noun |
| 0.01 | $x$ is a proper noun |
| −0.30 | There is an NP to the left of $x$ in $s$ |
| −0.43 | $20 \text{ words} < len(s)$ |
| −0.61 | $r$ matches V from Table 1 |
| −0.65 | There is a preposition to the left of $x$ in $s$ |
| −0.81 | There is an NP to the right of $y$ in $s$ |
| −0.93 | Cood. conjunction to the left of $r$ in $s$ |

### 4.2 Proposed Method: CTID

In this study, we develop features that help identify *competent* and *incompetent* triples in a triple set extracted from unstructured web text by the OpenIE system. The overall architecture and workflow of CTID are shown in Fig. 2. Here, a set of *reference texts* $R_T$ is used for the KG. $R_T$ refers to the context $c$ of the information for the KG. For each *reference text* $r_t$, we collect a set of relevant *sentence texts* $S_T$ extracted from the web in an unstructured text format to create triples. We then use OpenIE system to extract triples for each *sentence text* $s$ and create a *triple set* $\tau$ for each *reference text* $r_t$. To identify *competent* triples from the *triple set* $\tau$, we propose two types of feature, namely syntax- and semantic-based features. For each triple, we apply the proposed features and generate semantic and syntactic feature set. We then create a supervised machine learning model using the proposed features. The final output of this model is used to classify a triple as *competent* or *incompetent*. The proposed features are described in detail below.

#### 4.2.1 Syntax-Based Features

Syntax-based features mainly deal with the syntax of each triple. Here, we propose four syntax-based features (see Table 3). F1 is the confidence value obtained from the OpenIE system and F2, F3, and F4 are used to measure similarity using the Dice-coefficient.

Here, as a feature, we use confidence value of OpenIE system under each triple. Although it is not obvious that triples with high confidence values are always *competent*, we hypothesize that in combination with other features, F1 can improve the performance of the model.

Every triple has three parts [head $h$; relation $r$; tail $t$]. Here, F2, F3, and F4 measure the similarity of each part of the triple with the corresponding sentence $s$, where $s \in S_T$. F2 is related to the head part, F3 is related to the relation

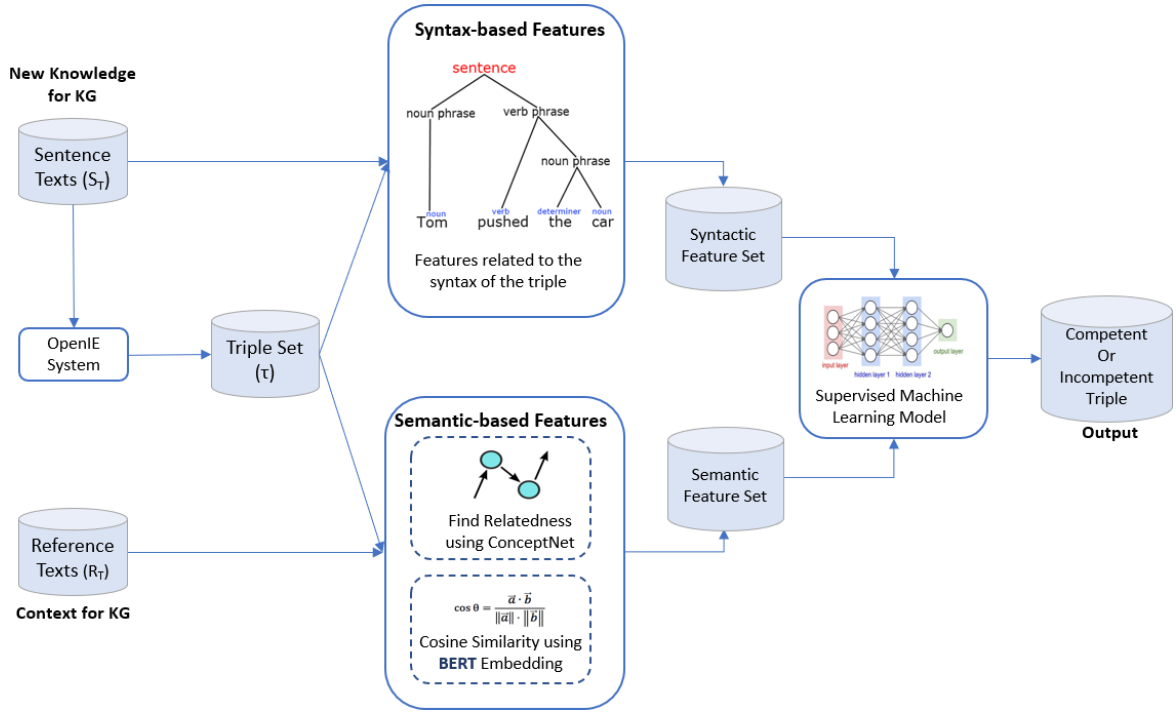



**Fig. 2** Overview of CTID model for identifying competent triples from unstructured text. For the extracted triple set $\tau$, the proposed syntax- and semantic-based features are prepared separately. Then, a supervised model is applied to classify the triples.

**Table 3** Proposed features

| No | Features |
|---|---|
| F1 | Confidence value from OpenIE Syntem |
| F2 | Sentence similarity between $s$ and $h$ uisng dice_coefficient |
| F3 | Sentence similarity between $s$ and $r$ uisng dice_coefficient |
| F4 | Sentence similarity between $s$ and $t$ uisng dice_coefficient |

part, and F4 is related to the tail part of each triple. These features are fully independent. Therefore, it is not necessary to maintain any order to calculate Dice-coefficient using these features.

The OpenIE system performs well for simple sentence patterns. For complex sentence patterns, it sometimes identifies only some part of the information contained in the input sentence. Therefore, a similarity measure can be used as a weight for the information extracted by the OpenIE system from sentence $s$. To find this similarity, we calculate the Dice-coefficient for each part of the triple and the corresponding sentence $s$. Equation (1) is used to calculate the Dice-coefficient between two sets $X$ and $Y$, where X is the set of terms in sentence $s$ and Y is the set of terms of the head part $h$ or relation part $r$, or tail part $t$ of a triple.

$$Dice\_coefficient(X, Y) = \frac{2|X \cap Y|}{|X| + |Y|} \quad (1)$$

Figure 3 explains features F2, F3, and F4 using two examples. In Example 1, the sentence $s$ is "Barack Obama was born in Hawaii." and the extracted triple is [Barack Obama(h); born in(r); Hawaii(t)]. For each part of the triple, we calculate the Dice-coefficient to find the similarity with

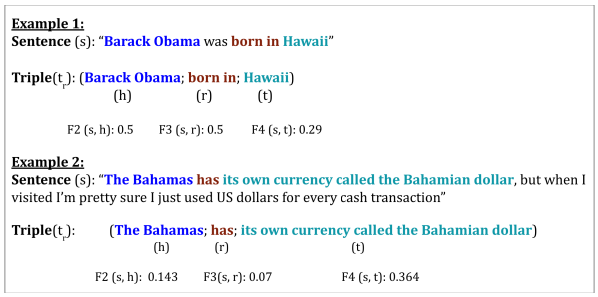



**Fig. 3** Description of proposed features F2, F3, and F4. Please refer to Sect. 4.2.1 for details.

the sentence. For this example, F2 is 0.5, F3 is 0.5 and F4 is 0.29. In Example 2, the structure of sentence $s$ is complex and the triple extracted by OpenIE does not cover the full-sentence pattern. For this example, F2 is 0.143, F3 is 0.07, and F4 is 0.364.

With these four proposed features, we also incorporate the features from the ReVerb system (Table 2). Those features are also independent. Since one feature value cannot dominate to classify the triple set, we identify competent triples by using all of the features in Table 2 and four proposed features.

### 4.2.2 Semantic-Based Features

Semantic-based features help to measure the semantic relatedness of an extracted triple with the corresponding *reference text* ($R_T$). For example, if the *reference text* ($R_T$)

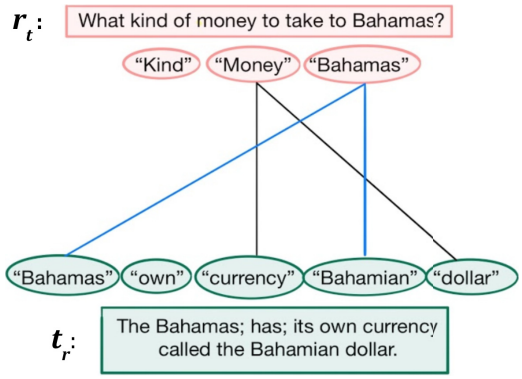



**Fig. 4** Token-based Semantic relatedness measure based on ConceptNet. "Money", "currency", and "dollar" are semantically related here.

refers to *"Birthplace of Barack Obama"*, then the competent triple has to be related to this context. Here, we propose two semantic-based features, namely a semantic relatedness measure that uses *ConceptNet*[†] (commonly used to compute semantic similarity) and a cosine similarity that uses BERT embedding [25] (a state-of-the-art model for natural language processing).

**Semantic Relatedness Measure based on ConceptNet:** We utilize ConceptNet to measure the semantic relatedness between each triple $t_r$ ($t_r \in \tau$) and the *reference text* $r_t$ ($r_t \in R_T$). ConceptNet is a widely used semantic network that helps computers understand the meanings of words. The latest version of ConceptNet covers a wide range of vocabulary for measuring semantic relatedness. Here, we measure word-level relatedness by employing the related word list from ConceptNet 5. This measure is easily interpretable for finding the semantic relation between reference text $r_t$ and extracted triple $t_r$.

We focus on words that define the meaning of the text. Therefore, we apply natural language processing techniques to tokenize the reference text $r_t$ and the relevant triple $t_r$. Here, we apply basic tokenization with POS-tag identification. For this purpose, we use spaCy[††], an open-source software library for advanced natural language processing. Here, we use the spaCy stop word list to remove stop words from both token lists. Then, we apply the spaCy lemmatizer to lemmatize the rest of the tokens of each list. ConceptNet is then applied to each remaining token to collect the top $N$ related words and create two related word lists, $W_R$ and $W_T$, by removing all duplicates. We then calculate the number of matches in these two lists using Eq. (2).

$$Semantic\_Relatedness\_Measure = W_R \cap W_T \tag{2}$$

This measure represents the relatedness between the reference text and the relevant triple.

Figure 4 shows an example of the semantic similarity measure. Here, reference text $r_t$ is "What kind of money to take to Bahamas?" and the relevant triple is $t_r$. After removing the stop words, we obtained two token lists, namely ("kind", "money", "Bahamas") from reference text $r_t$ and ("Bahamas", "own", "currency", "Bahamian", "dollar") from relevant triple $t_r$. For each token $x$, we collect the related word list. Here, $Rel(x)$ refers to the list of related words for a token. For example, $Rel(money)$ = *["bank", "wallet", "currency", "bill", "dollar", "account", ....]*, $Rel(dollar)$ = *["money", "currency", "bill", "cent", "price", ...]*, and $Rel(currency)$ = *["dollar", "money", "coin", "bill", "tax", ...]*. We then create two separate related word lists for reference text $r_t$ and relevant triple $t_r$ without any duplicates and calculate the number of matches. For example, in Fig. 4, we get common words for $Rel(money)$, $Rel(dollar)$, and $Rel(currency)$. They represent the semantic relatedness among each other. This example shows that ConceptNet can be used to measure the semantic relatedness between reference text $r_t$ and relevant triple $t_r$. More common words indicate a closer relation.

**Cosine Similarity based on BERT:** This feature is used to determine the similarity between triples and the reference text based on the cosine value. We utilize BERT [25] embedding to find a word vector for each token. Other popular word embedding models such as skip-gram [26], CBOW [26], and GLOVE [27] are context-free. This means that for "river bank" and "bank account", the models give the same embedding vector for "bank" despite the different meanings. In contrast, BERT embedding is contextual, which means that it can generate different representations based on meaning. The pretrained model covers a relatively wide range of sentences. To understand the actual meaning of natural text, this type of representation is essential. Therefore, we apply BERT [25] embedding to each reference text $r_t$ and each triple $t_r$. We then calculate the cosine similarity between each pair of reference text tokens and relevant triple tokens. If the tokens are the same, the feature value is set to zero because an identical token does not add any new information. We focus on the similarity between different tokens to determine the semantic relatedness between each relevant triple $t_r$ and reference text $r_t$. We use Eq. (3) to calculate the similarity measure.

$$Similarity\_Measure = \sum_{i=1}^{a}\sum_{j=1}^{b} f(x_i, y_j) \tag{3}$$

where,

$$f(x,y) = \begin{cases} cos\_sim(x,y) & \text{if } cos\_sim(x,y) \geq T_h \text{ and } x \neq y \\ 0 & \text{otherwise} \end{cases}$$

$$cos\_sim(x,y) = \frac{x \cdot y}{||x||\,||y||}$$

Here, $a$ denotes the length of reference text tokens, $b$ denotes the length of relevant triple tokens, $x_i$ denotes the embedding vector of $i^{th}$ token of reference text $r_t$, $y_j$ denotes the embedding vector of $j^{th}$ token of relevant triple $t_r$, and

[†]http://conceptnet.io/
[††]https://spacy.io/

$T_h$ denotes the threshold value.

We add a threshold value, $T_h$, for the cosine similarity. If the similarity of a pair is greater than or equal to $T_h$, we add the pair to the feature vector.

#### 4.2.3 Supervised Machine Learning Model

After calculating the syntax- and semantic-based features, we simply concatenate these features for our CTID model. We also concatenate the features from Table 2 to utilize the syntactic and lexical constraint mechanisms used in the ReVerb system. We then apply a supervised learning model to train our model. Here, we apply neural-network-based settings for our CTID (note that any supervised learning method can be used). The aim of this model is to classify the input triples either as *competent* or *incompetent*.

## 5. Experiment

### 5.1 Dataset

To evaluate the proposed features, we need a dataset that consists of triples with reference text. For such a dataset, which does not yet exist, a lot of human effort and domain expertise would be required to annotate each triple. Therefore, we utilize the question-answer dataset WebQuestionsSP [28]. In this dataset, questions are generated based on Freebase [3], which is a large collaborative KB. In this dataset, the answer entity is given for the questions of the training set. There are 3098 questions in the training set.

In our experiment, we used the 3098 questions as our reference text. We collected natural text data and then extracted triples using the OpenIE system. Each step of the dataset creation process is described below.

#### 5.1.1 Sentence Acquisition and Triple Generation

Here, we explain the extraction of text for each reference. In this experiment, a set of questions was considered to be a set of reference texts $R_T$. Using each question as a search query, we extracted corresponding snippet texts using the Google search engine. We employed the Google API Client and extract the top 10 answer snippets for each question, as shown in Fig. 5. We collected a total of 30980 snippets from the 3098 questions.

After collecting snippets, we extracted sentences from the snippets using text processing, as shown in Fig. 5. Because snippets do not always contain a complete sentence (ended by a full stop mark), we removed incomplete sentences (those not ended by a full stop mark). From the 30980 snippets, we obtained a total of 44440 sentences, which were used as the input for the OpenIE [7] system for generating triples. We used OpenIE v4, which is a combination of SRLIE [29] and RelNoun [30]. Table 4 shows an example of triple generation using OpenIE v4.

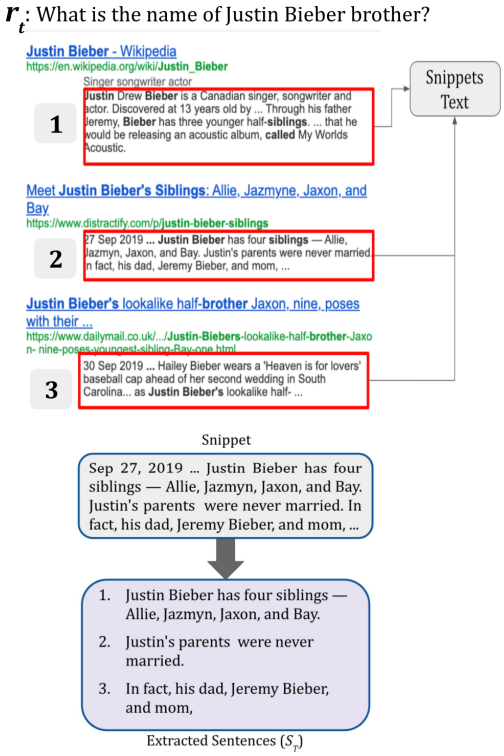



**Fig. 5** Sentence extracted from Google search results. For each reference text, the top 10 relevant snippets are first extracted. The sentences are then separated using text processing.

**Table 4** Example of triple generation using OpenIE v4

| | |
|---|---|
| Input Sentence | John ran down the road to fetch a pail of water. |
| Output of OpenIE v4 | 0.86 (John; ran; down the road; to fetch a pail of water) |
| | 0.82 John ran:(John; ran down the road to fetch; a pail of water) |

#### 5.1.2 Noise Removal

We removed noise related to the sentence pattern and noise related to the triple. These types of noise are generated due to the limitations of the OpenIE system. For some sentences, the OpenIE system cannot extract any triple. Therefore, we need to remove sentences that have no triples. In addition, some generated triples only have two parts, rather than three. Hence, we also need to remove incomplete triples from the extracted triple set $\tau$.

#### 5.1.3 Data Annotation

We divided all triples into two classes, namely *incompetent* and *competent*. As mentioned in Sect. 1, competent triples can contribute new information to a KG whereas incompetent triples cannot. We utilized questions from WebQuestionsSP with the answer entity for our experiment.

**Algorithm 1** Data Annotation Procedure

**Input** All Triple set $T$={$(h, r, t)$}, Question set $Q$, Answer entity set $A$

**initialize** $CompetentTripleSet$ = []
  $IncompetentTripleSet$ = []
**for** each question $q \in Q$ **do**
  $token_q \leftarrow$ tokenize $(q)$ ▷ Tokenize the Question text
  $token_q \leftarrow remove\_stop\_words(token_q)$
  $relWord_q \leftarrow$ related_word$(token_q)$ ▷ From ConceptNet
  **for** each triple $t_q \in T$ **do** ▷ Related to question $q$
    $token_{t_q} \leftarrow$ tokenize $(t_q)$ ▷ Tokenize the triple text
    $token_{t_q} \leftarrow remove\_stop\_words(token_{t_q})$
    **if** $a_q \in token_{t_q}$ **then** ▷ $a_q \in A$
      $relWord_{t_q} \leftarrow$ related_word$(token_{t_q})$
      $p \leftarrow relWord_{t_q} \cap relWord_q$
      **if** $length(p) \geq 1$ **then**
        $CompetentTripleSet \leftarrow t_q$
      **else**
        $IncompetentTripleSet \leftarrow t_q$
      **end if**
    **else**
      $IncompetentTripleSet \leftarrow t_q$
    **end if**
  **end for**
**end for**

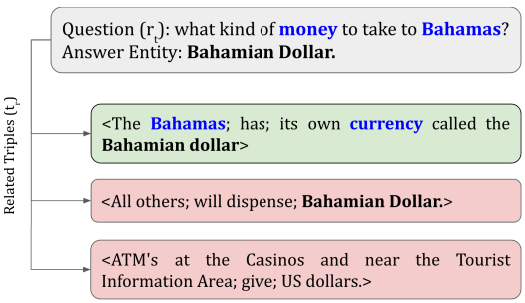



**Fig. 6** Example of data annotation. Green and red boxes respectively represent competent and incompetent triples.

To annotate the extracted triples, we utilized the answer entity. We propose a procedure for automatically annotating extracted triples.

Algorithm 1 describes the procedure of our data annotation. We first tokenize the triple as well as the corresponding question and then remove all stop words. We then check whether the token list of the triple contains the answer entity. If it does not, we label the triple as *incompetent* because a triple without the answer entity has no possibility of becoming a relevant triple of a question. A triple that contains the answer entity has a possibility of becoming a relevant triple but it is not always obvious if it will. Here, we measure the semantic relatedness of a triple with the reference text using ConceptNet. If the triple is semantically related to the reference, we label it as *competent*. We collect related words for each token. If we find some common words between the triple tokens and question tokens, we label the triple as *competent*; otherwise, we label it as *incompetent*.

Figure 6 shows an example of our data annotation procedure. In this example, for simplicity, we use the full triple and question (i.e., stop words are not removed). For a given question, we have three triples. We can see that the first two triples contain the answer entity. We then measure the semantic relatedness between these triples and the question. The first triple is semantically related. For example, the question token *"money"* is semantically related to the first triple tokens *"currency" and "dollar"*. Therefore, we label the first triple as competent and other two triples as incompetent. In Fig. 6, green indicates a competent triple and red indicates an incompetent triple.

**Table 5** Dataset summary

| | |
|---|---|
| Number of Questions | 3098 |
| Number of Extracted Snippets | 30980 |
| Total number of Sentences | 44440 |
| Total number of generated triples | 89179 |
| Total Number of Labeled Triples | 61500 |
| Number of Competent Triples | 1142 |

## 5.2 Experimental Settings

In this study, we propose two types of feature for identifying *competent* and *incompetent* triples in natural text data. As mentioned earlier, to evaluate these features, any supervised method can be used. Here, to evaluate these features, we use a neural-network-based model with two hidden layers. Details of the model's optimization are given in Sect. 5.2.1. Table 5 shows a summary of the dataset used in this experiment. There are 1142 competent triples in total. For the validation, we manually check the triples after annotation.

### 5.2.1 Model Optimization

We used a neural-network-based model with two hidden layers. Each layer was densely connected. The number of neurons in the first and second hidden layers was 300 and 100, respectively. We also evaluated our model using 10-fold cross-validation. For model optimization, we used the binary cross-entropy loss function with the stochastic gradient descent optimizer. For each hidden layer, we used the rectified linear unit (ReLU) activation function, and for the output layer, we used sigmoid activation function.

### 5.2.2 Evaluation Measures

For the evaluation, the standard information extraction measures (i.e., precision, recall, F1 score, accuracy) were applied. These measures were calculated as follows:

- *Precision:* Precision $P$ specifies the correct amount of information retrieved. Here, our main focus is to identify competent triples. Therefore, this measure refers to the proportion of correct triples assigned to the competent class that are actually members of this class. It is calculated using Eq. (4).

$$P = \frac{\text{Truly Identified Competent Triples}}{\text{Total Competent Triples Identified}} \tag{4}$$

- *Recall:* Recall $R$ represents the degree of correct infor-

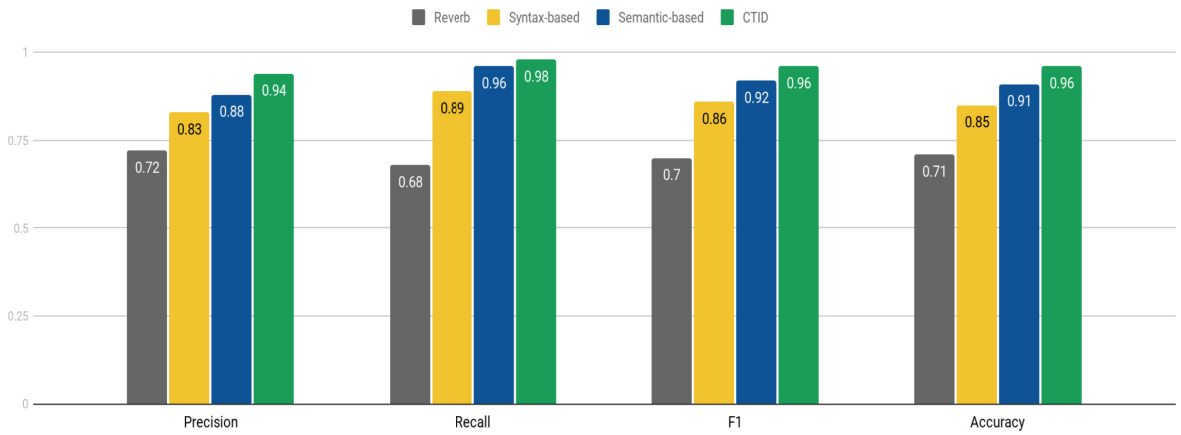



**Fig. 7** Experimental Result. Comparing the performance of proposed features.

mation retrieved. Therefore, it is the proportion of competent triples that the system assigns to this class. It is calculated using Eq. (5).

$$R = \frac{\text{Truly Identified Competent Triples}}{\text{Total Competent Triples}} \tag{5}$$

- *F1:* F1 score is the harmonic mean of precision $P$ and recall $R$. It is calculated using Eq. (6).

$$F1 = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

- *Accuracy* Accuracy $A$ is the most intuitive performance measure of a classifier. It is the ratio of correctly predicted observations to the total number of observations. Therefore, it is the proportion of correctly identified triples. It is calculated using Eq. (7).

$$A = \frac{\text{Truly Identified Triples}}{\text{Total Number of Triples}} \tag{7}$$

#### 5.2.3 Baseline Model

To design our baseline model, we utilized the features in the ReVerb system [17] (see Sect. 4.1 for details). We applied the same neural-network-based settings for designing the baseline model. Here, we compare the proposed features with the baseline features.

### 5.3 Experimental Results

Figure 7 shows the results of our experiment. To evaluate the effectiveness of our CTID model, we compare our model with the ReVerb system features. The ReVerb system is mainly focused on syntax-based features, and thus Figure 7 shows that with the combination of the proposed features and the ReVerb features, our model achieves about 20% better precision, 30% better recall, 25% better F1 score, and 25% better accuracy compared to those for the ReVerb system.

We also conducted an ablation analysis. We proposed two types of feature, namely syntax- and semantic-based features, for our CTID model. We conducted analyses using these features separately. Figure 7 shows the results of these analyses. Using only the syntax-based features resulted in better performance compared to that of the ReVerb system in terms of all evaluation measures. Using only the semantic-based features resulted better performance compared to that of the ReVerb system and syntax-based features. Therefore, semantic-based features are more effective than syntax-based features. In our CTID model, by applying both syntax- and semantic-based features, we can achieve better results compared to those obtained with either feature type alone. Therefore, both types of feature are necessary for accurately identifying competent and incompetent triples.

With both syntax- and semantic-based features in our CTID model, our approach outperformed the baseline by 20%, which is a significant improvement as determined using the $t$-test at level 0.95. Therefore, the CTID model is effective in identifying competent and incompetent triples.

## 6. Discussion

Although the CTID model had the highest precision in the evaluation, some *competent* triples were not identified by this model. We investigate these missed triples to assess the effectiveness of the CTID model. Table 6 shows some of the input triples and the model output with correct answers.

The output shows that triples, which identified as *incompetent* by the CTID model, have some semantic relation with the reference text and also contained answer entity. But these triples do not contain the primary question entity. For example, for the question “What are the primary language of France?”, the answer entity is “French”, which is present in the corresponding triple. However, the primary question entity “France” is not present in that triple. This is the limitation of our data annotation procedure. Because the annotation is automatic, these types of triples are annotated as *competent*. Despite this limitation, it may be possible to

**Table 6** Output examples of the CTID model

| Triples Information | Model Output | Correct Answer |
|---|---|---|
| Reference Text: "ques": "what is the **currency** name of **Brazil**?", "ans": "**Brazilian real**" "triple": "The **Brazilian real**", "is", "the official **currency** of **Brazil**" | Competent | Competent |
| Reference Text: "ques": "what are the primary **language**s of France?", "ans": "**French**" "triple": "**French**, the official **language**", "is", "the first **language** of 88% of the population" | Incompetent | Competent |
| Reference Text: "ques": "what **political** party was Hitler the leader of?" "ans": "Nazi Party","**German Workers' Party**" "triple": "a fledgling **political** organization", "called", "the **German Workers' Party**" | Incompetent | Competent |
| Reference Text: "ques": "what religion does Tom Cruise follow?" "ans": "**Scientology**", "Catholicism" "triple": "**Scientology**", "is", "a body of religious beliefs and practices first described in 1950" | Incompetent | Competent |
| Reference Text: "ques": "what disease did abe lincoln have? "ans": "Strabismus", "Smallpox", "**Marfan syndrome**" "triple": "**Marfan syndrome**", "is", "an autosomal dominant disorder" | Incompetent | Competent |
| Reference Text: "ques": "what **countries share border**s with France?", "ans": "Belgium","**Germany**", "Italy", "Luxembourg", "Monaco", "Spain", "Switzerland", "United Kingdom", "Andorra" "triple": "**Germany**", "**share**s", "a land **border** with nine other **countries**" | Incompetent | Competent |

utilize the automated annotation procedure to assist human-level annotation.

In our experiment, besides the neural network parameters, we also used two additional parameters, namely $N$ and $T_h$. $N$ is the number of related words extracted from ConceptNet for each token. Related words can include those in other languages. Here, we only consider English words. Most English words are at the top of the related word list. Hence, this parameter does not need to be tuned. $T_h$ is the threshold value, which is used for the similarity measure in Eq. (3). The different threshold values could affect the system's performance. However, the results were not significantly different when we experimented on the threshold between 0.6–0.8. We found that the system is robust to this parameter in some range based on the experimental results.

To measure semantic relatedness, we use the ConceptNet semantic network and BERT embedding, both of which cover a wide range of vocabulary. Despite this, considering the open-world assumption, there may have unseen words, which are out of the vocabularies of ConceptNet and BERT. In this study, our main target is introducing new knowledge to a KG that is not available in existing KGs, so the vocabulary limitation was not considered.

## 7. Conclusion

In this paper, we proposed syntax- and semantic-based features for identifying competent triples in unstructured natural text data. We use the OpenIE system to extract triple format data from natural text. Our features can identify competent triples from the triple set. These triples have a low chance of adding noise to existing KGs. We also proposed an automatic annotation procedure that does not require domain knowledge and thus reduces the need for human intervention. This procedure can be used for any domain. The experimental results show that both syntax- and semantic-based features outperform the baseline features. These results confirm that the proposed CTID model can identify competent triples. In the future, we will add these triples to complete existing KGs and also try to alleviate the limitation of the annotation procedure.

## Acknowledgements

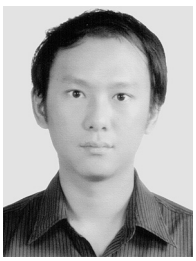
This work was partially supported by the New Energy and Industrial Technology Development Organization (NEDO).

### References

[1] D. Vrandečić and M. Krötzsch, "Wikidata: A Free Collaborative Knowledge Base," Communications of the ACM, vol.57, no.10, pp.78–85, 2014.
[2] J. Hoffart, F.M. Suchanek, K. Berberich, and G. Weikum, "YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia," Artificial Intelligence, vol.194, p.28–61, 2013.
[3] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A Collaboratively Created Graph Database For Structuring Human Knowledge," in Proceedings of the ACM SIGMOD International Conference on Management of Data, pp.1247–1250, 2008.
[4] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge Graph Embedding by Translating on Hyperplanes," Proc. 28th AAAI Conference on Artificial Intelligence, pp.1112–1119, 2014.
[5] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," Proc. 33rd International Conference on International Conference on Machine Learning, pp.2071–2080, 2016.
[6] M. Tanwar, R. Duggal, and S.K. Khatri, "Unravelling unstructured data: A wealth of information in big data," in Proceddings of the 4th International Conference on Reliability, Infocom Technologies and

Optimization, pp.1–6, 2015.
[7] M. Banko, M. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, "Open Information Extraction from the Web," Proc. 20th International Joint Conference on Artifical Intelligence, p.2670–2676, 2007.
[8] L.D. Corro and R. Gemulla, "ClausIE: Clause-based Open Information Extraction," in Proceedings of the 22nd international conference on World Wide Web, pp.355–366, 2013.
[9] K. Gashteovski, R. Gemulla, and L.d. Corro, "MinIE: Minimizing Facts in Open Information Extraction," in Proceedings of the Conference on Empirical Methods in Natural Language Processing, ACL, pp.2630–2640, 2017.
[10] D. Liben-Nowell and J. Kleinberg, "The Link-prediction Problem for Social Networks," Journal of the American Society for Information Science and Technology, vol.58, no.7, pp.1019–1031, 2007.
[11] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning Entity and Relation Embeddings for Knowledge Graph Completion," in Proceedings of the 29th AAAI Conference on Artificial Intelligence, pp.2181–2187, 2015.
[12] T. Ebisu and R. Ichise, "TorusE: Knowledge graph embedding on a lie group," Proc. AAAI Conference on Artificial Intelligence, pp.1819–1826, 2018.
[13] L.A. Galárraga, C. Teflioudi, K. Hose, and F. Suchanek, "AMIE: Association Rule Mining under Incomplete Evidence in Ontological Knowledge Bases," in Proceedings of the 22nd International Conference on World Wide Web, pp.413–422, 2013.
[14] T. Ebisu and R. Ichise, "Graph Pattern Entity Ranking Model for Knowledge Graph Completion," in Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp.988–997, 2019.
[15] H. Paulheim and C. Bizer, "Improving the Quality of Linked Data using Statistical Distributions," International Journal on Semantic Web and Information Systems, vol.10, no.2, pp.63–86, 2014.
[16] L. Zhao, R.F. Munne, N. Kertkeidkachorn, and R. Ichise, "Missing RDF Triples Detection and Correction in Knowledge Graphs," in Proceedings of the 7th Joint International Semantic Technology Conference, p.164–180, Springer, 2017.
[17] A. Fader, S. Soderland, and O. Etzioni, "Identifying Relations for Open Information Extraction," Proc. Conference on Empirical Methods in Natural Language Processing, pp.1535–1545, ACL, 2011.
[18] M. Schmitz, R. Bart, S. Soderland, O. Etzioni, et al., "Open Language Learning for Information Extraction," Proc. Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp.523–534, 2012.
[19] N. Kertkeidkachorn and R. Ichise, "T2KG: An End-to-End System for Creating Knowledge Graph from Unstructured Text," Proc. AAAI Workshop on Knowledge-based Techniques for Problem Solving and Reasoning, pp.743–749, 2017.
[20] I. Augenstein, S. Padó, and S. Rudolph, "LODifier: Generating Linked Data from Unstructured Text," in Proceedings of the 9th International Conference on Semantic Web: Research and Applications, pp.210–224, Springer, 2012.
[21] V. Kríž, B. Hladká, M. Nečaský, and T. Knap, "Data Extraction Using NLP Techniques and Its Transformation to Linked Data," in Proceedings of the Mexican International Conference on Artificial Intelligence, pp.113–124, Springer, 2014.
[22] H. Alani, S. Kim, D.E. Millard, M.J. Weal, W. Hall, P.H. Lewis, and N.R. Shadbolt, "Automatic Ontology-based Knowledge Extraction from Web Documents," IEEE Intell. Syst., vol.18, no.1, pp.14–21, 2003.
[23] A. Yates, M. Cafarella, M. Banko, O. Etzioni, M. Broadhead, and S. Soderland, "TextRunner: Open Information Extraction on the Web," in Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp.25–26, ACL, 2007.
[24] F. Wu and D.S. Weld, "Open Information Extraction Using Wikipedia," Proc. 48th Annual Meeting of the Association for Computational Linguistics, pp.118–127, 2010.
[25] J. Devlin, M.W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Proc. Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp.4171–4186, 2019.
[26] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," Workshop Track Proceedings of the 1st International Conference on Learning Representations, 2013.
[27] J. Pennington, R. Socher, and C. Manning, "GloVe: Global Vectors for Word Representation," in Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp.1532–1543, 2014.
[28] W.-T. Yih, M. Richardson, C. Meek, M.-W. Chang, and J. Suh, "The Value of Semantic Parse Labeling for Knowledge Base Question Answering," in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pp.201–206, ACL, 2016.
[29] J. Christensen, Mausam, S. Soderland, and O. Etzioni, "An Analysis of Open Information Extraction based on Semantic Role Labeling," in Proceedings of the 6th International Conference on Knowledge Capture, pp.113–120, 2011.
[30] H. Pal and Mausam, "Demonyms and Compound Relational Nouns in Nominal Open IE," in Proceedings of the 5th Workshop on Automated Knowledge Base Construction, pp.35–39, 2016.

**Esrat Farjana** received her B.Sc degree in Computer Science and Engineering from Military Institute of Science and Technology (MIST), Dhaka, Bangladesh, in 2015. She is currently a Ph.D. candidate at The Graduate University for Advanced Studies, SOKENDAI, in Japan. Her research interests include information retrieval, machine learning, and natural language processing.



**Natthawut Kertkeidkachorn** received a Ph.D. degree in Informatics from The Graduate University for Advanced Studies, SOKENDAI, Japan in 2017. He is currently an assistant professor at the School of Information Science, Japan Advanced Institute of Science and Technology. His research interests include the Semantic Web, machine learning, and natural language processing.

**Ryutaro Ichise** received a Ph.D. degree in computer science from Tokyo Institute of Technology, Tokyo, Japan, in 2000. From 2001 to 2002, he was a visiting scholar at Stanford University. He is currently an associate professor in the Principles of Informatics Research Division at the National Institute of Informatics in Japan. His research interests include the Semantic Web, machine learning, and data mining.