Knowledge Extraction for Open Knowledge Graph under Open World Assumption

Esrat Farjana Rupu

Doctor of Philosophy

Department of Informatics

School of Multidisciplinary Sciences

The Graduate University for Advanced Studies, SOKENDAI

September 2023

Advisory Committee

1. Prof. Hideaki Takeda	National Institute of Informatics SOKENDAI
2. Prof. Ken Satoh	National Institute of Informatics SOKENDAI
3. Prof. Akiko Aizawa	National Institute of Informatics SOKENDAI
4. Asst. Prof. Saku Sugawara	National Institute of Informatics SOKENDAI
5. Prof. Ryutaro Ichise	Tokyo Institute of Technology National Institute of Informatics

Acknowledgments

I would like to express my sincere gratitude to my advisor Prof. Ryutaro Ichise for the continuous support of my doctoral study and research and for his patience, encouragement, and immense knowledge. Without his guidance and support, this thesis would not have been possible. I greatly appreciate all his contributions of time, ideas, and supervision to make my research experience productive and stimulating.

I would like to thank advisory committee members: Prof. Hideaki Takeda, Prof. Ken Satoh, Prof. Akiko Aizawa, and, Asst. Prof. Saku Sugawara for not only their insightful and constructive comments and encouragement but also for the fruitful suggestions that inspired me to broaden my research from various perspectives.

I appreciate my all lab members as well as all internship students, and anyone who encouraged me and provided thoughtful comments on my research.

Most importantly, I would like to express my heartfelt gratitude to my family members, who have been a constant source of love, concern, encouragement, and continuous support throughout my years of studying, researching, and writing this thesis. None of this would have been possible without them.

Lastly, I would like to thank The Graduate University for Advanced Studies, SOKENDAI, the National Institute of Informatics, and the Ministry of Education, Culture, Sports, Science and Technology (MEXT) for their educational and financial support during my doctoral studies.

Abstract

A Knowledge Graph (KG) is a semantic network representing real-world entities - i.e., objects, events, situations, or concepts. It illustrates the relationship between them. The information is usually stored in a graph database and visualized as a graph structure. KGs are widely used for various AI-related tasks, such as web search, question-answering, entity linking, semantic parsing, named entity disambiguation to information extraction, and question answering. Several efforts have been made to develop knowledge graphs in general and specific domains such as DBpedia, YAGO, LinkedGeoData, and Wikidata. However, with the advent of the internet, new knowledge is emerging every day which makes existing KGs incomplete. In addition, it is very difficult to add new knowledge to the existing structure-based KGs. Most of the KGs are made based on the close world assumption where all entities and relations are known in advance. Therefore, adding new entity information in the existing settings is difficult. In contrast with the closed-world assumption, all entities and relations are not known in advance in the open-world assumption, and most of the knowledge is available in natural text format which makes it difficult to extract information for KG. However, OpenIE tools can extract triple-format data from a given text without any prespecified vocabulary, but it includes lots of noise, making it inconsistent. There is no system available to extract suitable information from the extracted triples by OpenIE tools. In this thesis, we develop a model CTID (Competent Triple Identification) to find competent triples from the extracted triple set by the OpenIE tool.

In our CTID model, we develop two types of features, namely syntax- and semantic-based features, to identify competent triples from a given triple set. For each triple, we apply the proposed features and generate semantic and syntactic feature sets. 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. In the syntactic feature set, we have used a total of 17 features. These features extract those triples which are syntactically suitable for KG. On the other hand, in semantic-based features, we apply cosine similarity using BERT embedding and utilize conceptNet to identify each triple relatedness. As there is no suitable dataset for this task, by utilizing the QA dataset we build a new dataset for this task and annotate the triples. We apply a supervised machine learning algorithm to evaluate our proposed feature set and analyze the result from different perspectives. Using different types of analysis, we can see that our model can classify competent and incompetent triples with a good result using two types of features.

Extracting information from natural text is very complex. Although OpenIE tools can extract triple-format information from a given text, it is not suitable for KG because of its noisy data. To fill this gap, we design our CTID model which can extract competent triples from a given triple set that can be utilized for KGs. With this study, we aim to identify competent triples from lots of noisy information. Competent triples identified by the CTID model can be utilized for existing KGs completion tasks and to create an open knowledge graph. Therefore, the result from this model can be used in future directions.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

In recent years, the amount of available data has dramatically increased due to the growth of the Internet. Since the invention of the World Wide Web [1] by Tim Berners-Lee, the data on the web has been generated and published continuously. The availability of data on the web is becoming more and more extremely rich. As a result, we can acquire fruitful knowledge from the data on the web. Such knowledge can obviously assist both a human and a machine to make a decision. For example, a human uses the knowledge to comprehend his/her interest as the prerequisite background or a search engine uses the knowledge as the prior knowledge to retrieve a relevant document. In order to collect and utilize the knowledge efficiently and effectively, a suitable technology is required. Traditionally, a knowledge base is a technology used to manipulate and store knowledge through a computer system. Currently, a modern knowledge base has become popularly known as Knowledge Graph [2].

Knowledge Graph (KG) is a structured knowledge base, which stores knowledge in

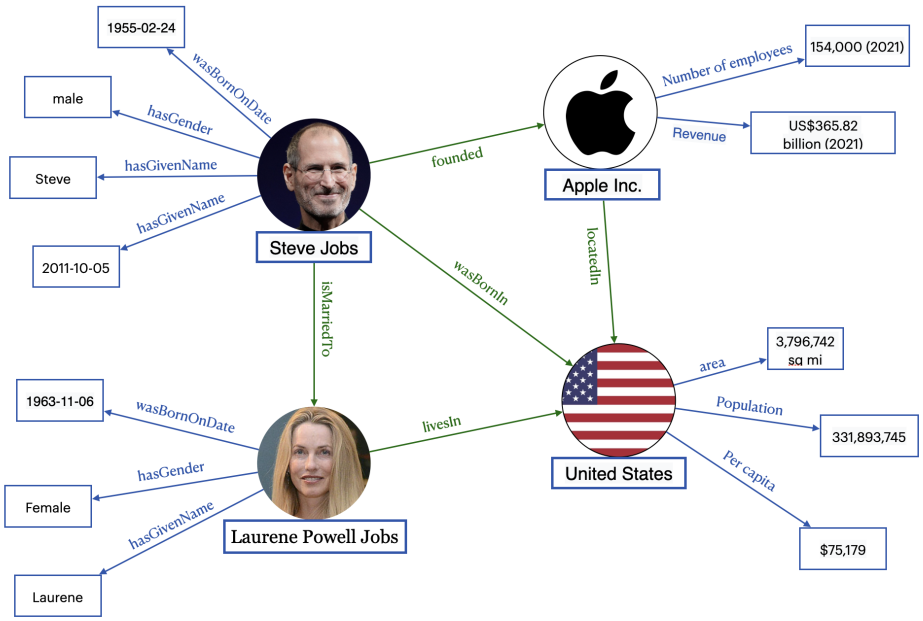

Figure 1.1: Example of Knowledge Graph

the form of real-world entities and their relationships. The term Knowledge Graph became widely known because of the release of Google's Knowledge Graph in 2012 [3]. Recently, the KG term gradually gained the attention of many researchers, especially in the semantic web community, because the main concept behind KG is the Linked Data concept [4], which is the major technology for the semantic web.

Knowledge Graph (KG) is modeled by a graph structure and facts are mainly represented in triple format. A triple consists of a subject, a relation, and an object where the relation indicates the relationship between an entity as the subject and the object. A relational triple can be denoted as (h, r, t) where h and t are the head entity and the tail entity, respectively, and r is the relation between the h and t. Figure 1.1 shows the example of a typical KG scenario.

Some of the well-known KGs are DBpedia [10], Freebase [11], and YAGO [12]. Such KGs play an important role in many advanced applications such as question and answering systems, quiz generation systems, search engines and etc [13]. For example, question and answering systems [14, 15], have used KGs as the prior knowledge to answer a specific question given by a user. A quiz generation system uses the

knowledge in KG to formulate a question and choice for a user to answer [16]. Search engines [3, 17] use KGs to understand the concept of search keywords to transform a text-based search engine into a semantic-based search engine, which can semantically understand a user's query. Apart from the open-domain KGs, several specific domain KGs such as Bio2RDF [18] and Neurocommons [19] have been used to support decisions in the life science domain applications. As a result, KGs become the prominent resource for many modern artificial intelligent systems.

1.2 Motivation

Since KGs play an important role in many modern applications as prior knowledge, many researchers, especially in areas of the semantic web and natural language processing, pay huge attention to constructing and populating knowledge into KGs. In recent years, there are existing KGs such as DBpedia, Freebase, YAGO and etc. However, new knowledge regularly emerges every day. Consequently, the current KGs gradually become incomplete and some knowledge of KGs may not be useful in the future. Considering the fact about the president of the United States in 2017 as an example, if KGs are not updated, the knowledge about the president of the United States provided by KGs is "Barack Obama". Nevertheless, in 2017 a new president of the United States "Donald Trump" is elected. The knowledge about the president of the United States in KGs becomes out of date. Consequently, when we search for the president of the United States, we could retrieve "Barack Obama" as a result, which becomes inappropriate in the current context. Another example is that suppose a new movie will release very soon, KGs might not be able to provide information about such a movie because knowledge about such information is missing. As a result, such KGs become not useful. Therefore, it is necessary to populate new knowledge to existing KGs in order to keep the existing KGs up to date.

For the knowledge graph completion task, an embedding-based model is commonly used in the KG completion task. Existing embedding-based KG completion methods such as TransE [6] and ComplEx [7] 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.

Generally, most of the new knowledge has been published as natural language text on the web and such a trend is dramatically growing faster than the growth of KGs [20]. Since such natural language text has been published on the web, we can access them easily. Nevertheless, natural language text has been traditionally treated as a string, which cannot be explicitly interpreted as any meaning or does not contain schemas or any links to any KGs. Moreover, due to language complexity, which relates to the structure of the language, it is not feasible for a machine to understand knowledge in natural language text directly. Furthermore, a publisher usually publishes natural language text by using his/her own vocabulary. It leads to the heterogeneous problem, where an identical real-world thing could be represented by many representations. Based on these reasons, a large amount of natural language text cannot be straightforwardly transformed into KGs and so is left as natural language text.

Furthermore, although a human can directly consume natural language text, a machine cannot make much use of such knowledge in the form of natural language text. The main reason is that a machine cannot understand a concept or a meaning in natural language text. Consequently, a machine loses an opportunity to use rich knowledge resources, which is left as natural language text. Due to the advantage of KGs, it is therefore essential to transforming natural language text to KGs so that a machine can also utilize the knowledge more efficiently and effectively.

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 [8]. 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) [9, 10, 11] system extracts 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.

1.3 Contribution

To address and solve the research problem above, we propose a complete model to solve that problem. 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.

1.4 Outline

We conclude this first chapter by outlining the structure of this dissertation. This dissertation is structured into five chapters. The remaining of our dissertation is organized as follows.

- **Chapter 2**

  This chapter presents the fundamentals of the knowledge graph and its concept and knowledge representation. Then, the background knowledge for knowledge graph construction. Later, we further discussed the natural language models that frequently uses in the KG alignment and enhancement task. We also discuss the OpenIE system to extract triples from natural text. Finally, we discussed closed and open-world assumptions before concluding the chapter.

- **Chapter 3**

  This chapter presents the details of the proposed CTID model for competent triple identification from natural text using the OpenIE tool. Here, our proposed features are discussed elaborately. In the CTID model, we proposed two types of feature sets namely syntax- and semantic-based features. In this chapter, we briefly explained these two types of feature sets and how these features contribute to extracting triples for the KGs. Next, we include the experiment conducted to evaluate the proposed features. Here, we utilize a QA dataset to create our dataset for the evaluation. We also include an ablation study and some limitations here. Finally, we discuss and summarize the CTID model.

- **Chapter 4**

  This chapter discusses the limitations and the achievement we accomplished by the proposed model using two different types of feature sets. Here, we include additional experiments which include human label annotation and compare the result which is built by using our annotation algorithm. We can understand the limitations of our annotation algorithm from this further analysis. In this chapter, we also discuss some additional parameters that are used in our proposed model and how to choose those parameters and what is the effect of those parameters. Here, we also include the assumptions of our research briefly.

- **Chapter 5**

  This chapter summarizes the thesis's contributions and outlines future work directions.

2

Fundamentals and Related Work

2.1 Knowledge Graph

Many studies use the term Knowledge Graph in many ways [12]. Although, in the previous chapter, we presented the background and necessity of Knowledge Graph and many related technical terms, their definitions are still not clearly clarified yet. We are therefore going to formally define the terms and give some further backgrounds and fundamentals for Knowledge Graph as follows.

A knowledge graph is a network of real-world objects, events, situations, or concepts and illustrates which are commonly referred to as entities and the relationship between them. This information is usually stored in a graph database and visualized as a graph structure, prompting the term knowledge “graph.”

2.1.1 Knowledge Graph Definition

Knowledge Graph is a graph-based knowledge base, which models knowledge between entities and relations [13]. Its definition can be formalized in the following definition.

Definition 1 *Knowledge Graph (KG): A Knowledge Graph KG = (V, E), where V is a set of vertices and E is a set of edges with a label. A vertex or a node in KG is an entity, while a labeled edge in KG is a relation.*

- ***Entity*** *is a vertex or a node in KG, which represents a unique real-world object. Note that, a description of the entity, referred as literal, can also be a node in KG.*
- ***Relation*** *is an edge with the label in KG, which expresses the relationship between entities or between an entity and its description.*

2.1.2 Knowledge Graph Representation

Knowledge Graph is represented by the Linked Data concept [13]. In the Linked Data concept, the Resource Description Framework (RDF) is a standard model for publishing Linked Data [14]. It uses to describe the information and their relations and also make data become interchangeable [14]. The specification of RDF is introduced by W3C Recommendation (RDF 1.1 Concepts and Abstract Syntax) [1]. In the RDF specification, the key concept is an RDF graph, which is a collection of RDF triples. An RDF triple consists of a subject, a predicate, and an object. A subject and an object are treated as a vertex, while a predicate is considered as a relation. The direction of the edge is used to identify which vertex is the subject and which vertex is the object. The edge of an RDF triple points out from the subject and points into the object. An RDF triple therefore can be viewed as a directed graph, which composes of two vertices and one directed edge, as shown in Figure 2.1.

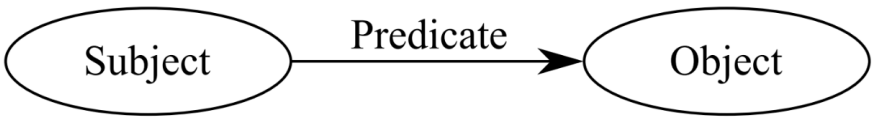

Figure 2.1: RDF Graph Representation [1]

Furthermore, the RDF specification [1] defines three kinds of resource representations: Internationalized Resource Identifier, literal and blank node. Such representations are used to describe an element of an RDF triple. Note that, based on our observation of many KGs, a blank node is usually ignored because the blank node does not provide any meaning and makes the representation of a KG become more complicated. Therefore, in our study, the blank node is not covered.

Due to the characteristic of an RDF triple as shown in Figure 2.1, a resource representation that can be used to describe each element of an RDF triple, is therefore dependent upon the position, which is a subject or a predicate or an object, of the element.

- **Subject**: The subject can be represented only by a URI. Since the subject is the entity. it needs to be identified as a unique resource, which is only represented by URI.

- **Predicate**: The predicate is also expressed by a URI. As shown in Figure 2.1, a predicate is an edge with its label. Different relations, therefore, can be expressed by different types of labeled edges.

- **Object**: The object can be a URI or literal. In contrast with the subject, an object is an information that fulfills the relation with its subject. Therefore, the object allows the representation as URI or Literal. If it is URI, it expresses the relation between entities, subjects, and objects. In the case of literal, it describes the detail for the subject, known as its description

2.2 Knowledge Graph Assumptions

Based on the survey [13], the interpretation assumptions are required so that knowledge of KGs can be understood. In KGs, the relationships between entities and their description are stored as knowledge, formally known as fact triples. It is obvious that a KG is incomplete. Therefore, non-existing triples in KGs have to be defined. Generally, there are two different assumptions: closed world assumption and open word assumption, which use to interpret the meaning of non-existing triples.

Table 2.1: Examples of RDF triples

subject	predicate	object
orgA-res:Smith	orgA-term:employedDate	"2017-06-01"^^xsd:date
orgA-res:Smith	orgA-term:name	"John Smith"
orgA-res:Smith	orgA-term:department	orgA-res:Global_Business
orgA-res:Smith	rdf:type	dbo:Person
orgA-res:Smith	owl:sameAs	orgB-res:Smith
orgB-res:Alice	dbo:spouse	orgB-res:Smith

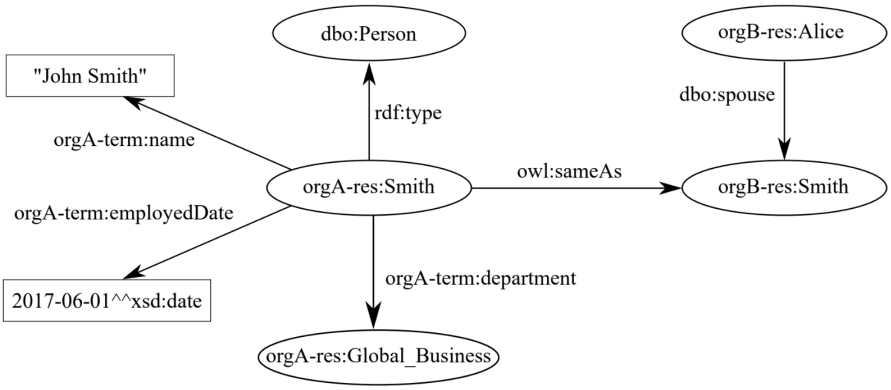

Figure 2.2: An Example of RDF Graph

- **Closed World Assumption:** Closed world assumption simply assumes that non-existing triples imply relationships between entities have not existed. In other words, it assumes that knowledge in a KG is already completed. Although this assumption reduces the complexity of the incomplete problem of a KG, its usage is extremely limited. Considering an extreme case shown in Table **??**, there is no relation between orgB-res:Alice and dbo:Person. Based upon this assumption, it concludes that "Alice is not a person". Based on the closed world assumption, it is correct; however, in fact, it is still not possible to conclude at this stage because of the lack of knowledge.

- **Open World Assumption:** Open world assumption supposes that non-existing triples cannot be interpreted and be treated as unknown. Considering the

same example with the closed world assumption, there is no relation between orgB-res:Alice and dbo:Person. Under the open world assumption, we cannot conclude “Alice is not a person”. Instead of that, we conclude that “Alice may or may be not a person”.

2.3 Knowledge Graph Construction

Knowledge Graph Construction is to collect knowledge and build a KG from such knowledge. The approach for the knowledge graph can roughly categorize into three approaches: 1) the manual approach, 2) the semi-automatic approach, and 3) the automatic approach. In the manual approach, a KG is manually created by mainly using human effort. In the semi-automatic approach, human efforts are put to craft rules or patterns so that a KG is automatically constructed. In the automatic approach, a KG is automatically generated by using various techniques, which reduces human intervention. The further details and well-known projects for each approach are listed as follows.

2.3.1 Manual Approach

In the manual approach, triples are manually gathered and integrated in order to build a KG. There are two popular methods: the curated method and the collaborative method, for the manual approach. In the curated method, a KG is built by a closed group of experts while in the collaborative method, a KG is crafted by an open community. Each method is presented in the following sub-sections.

2.3.1.1 Curated Method

The curated method usually aims to build a specialized KG for some specific purpose because this method requires a huge effort from humans to build a KG. Furthermore, in the curated method a KG is built by using a closed group of experts. Such a group of experts collect knowledge from a specific data resource and comply with such knowledge to produce the KG. Example projects for the curated method are present in the following list.

- **Cyc/OpenCyc [15]**
  Cyc/OpenCyc is one of the pioneer projects for constructing KG, formerly known as a knowledge base. The aim of the project is to establish, collect and assemble ontology and ontology of common sense knowledge in daily life so that an intelligent agent or system can utilize such knowledge in Cyc/OpenCyc for reasoning in target applications. In this project, more than a million axioms have been manually collected by Cycorp Inc., which is a company investing to create a large-scale knowledge base [15]. Here, we presented some usage of knowledge. Some examples of knowledge in this project are "every tree is a plant" and "every plant dies eventually" [16]. Such knowledge, referred to as rules, can be used for inferencing and reasoning in order to discover new knowledge. For example, if we know that the cherry blossom is a tree, based on the example rules, we can entail that the "cherry blossom dies eventually".

- **WordNet [17]**
  The WordNet project aims to construct a KG, which is also known as a lexical resource. In WordNet, a collection of lexicons and their semantic relationship are stored as knowledge. Fundamentally, there are six main semantic relationships between lexicons in the KG as follows [17].

  - **Synonymy** is the relationship that identifies the identical semantic relationship between lexicons. For example the lexicon "good" and the lexicon "well" holds the synonymy relation because their meanings are identical.
  - **Antonymy** is the relationship that describes the opposite semantic relationship between lexicons. For example, the lexicon "good" and the lexicon "bad" holds an antonymy relation because their meanings are opposite.
  - **Hyponymy and Hypernymy** are the relation that identifies subset and superset between lexicons respectively. Sometimes, these relationships refer to as is-A relations. For example "Dog" is a hypernym of "Animal", because every dog is a kind of animal. Here, Hyponymy and Hypernymy are considered as a semantic relation (is-A) because they are inverse of each other.

  - **Meronymy and Holonymy** are the relationship that describes the partOf and is-partOf relationship between lexicons respectively. For example, given the lexicon "door" is a meronym of "house", it can be interpreted as "door is a part of the house". Consequently, this relation frequently refers to as part of the relationship. Here, Meronymy and Holonymy are considered as one semantic relation because they are inverse of each other.
  - **Troponymy** is a relation that presents a co-occurrence between lexicons. For example, "to bite" is a troponym of "to eat" since the activity "to bite" is doing "to eat" in some manner.
  - **Entailment** is a relation that infers that one lexicon cause the result to the other one. For example, "to cry" is entailed by "to tears flow" because when crying, your tears must flow.

  WordNet is the linguistic domain lexicon resource, which contains many relationships between lexicons. Therefore, the construction process needs a group of experts in the linguistic domain to collect and build this KG so that the quality can be controlled. Recently, WordNet becomes the most valuable KG resource for the linguistic community. Therefore, it is widely used in many Natural Language Processing (NLP) applications [18].

- **UMLS [19]**
  The Unified Medical Language System (UMLS) project is also a specialized domain KG. It aims to construct the KG regarding the biomedical-related domain. Also, the project was developed by an expert group at the US National Library of Medicine [19]. In the UMLS project, more than ten million triples relationships among almost a million concepts are stored in the repository. Both ontology and taxonomy are integrated from various sources by the creator group.

As shown in the above projects [15, 17, 19], the curated method's main purpose is for the specific target for a specific purpose. Moreover, the limitation of this method is that it requires the curator in the group to collect, manipulate and update the knowledge in a KG. Although the curated method can create a small portion of knowledge for a specific purpose, it consumes a lot of resources such as time, and human effort. Specifically, in a large-scale project like Cyc/OpenCyc. The time estimation for completing the project

is more than 350 man-years (as estimated in 1986) [20]. However, new knowledge emerges very. As a result, we can not accurately estimate time.

2.3.1.2 Collaborative Method

The collaborative method aims to manually build a KG by people similar to the curated method. However, instead of a closed group of experts, the collaborative method allows an open community to collect knowledge and create a KG. This collaborative method is known as crowdsourcing [21]. In crowdsourcing, online communities play a significant role in the KG construction process. Conventionally, an existing platform is published online so that Internet users, who are interested in contributing to the project, can access and help the community to create a wide range of KGs. Two examples of the collaborative method project are presented as follows.

- **Freebase [22]**
  Freebase was a large collaborative knowledge base launched in 2007 as well. Google took it over in 2010 [22]. It was used as the open core of the Google Knowledge Graph project and has been attracted by many use cases outside Google. Due to the success of Wikidata, Google decided to close Freebase in 2014 and help with the migration of the content to Wikidata [23]. Freebase is built on the notions of objects, facts, types, and properties. Each Freebase object has a stable identifier called a “mid” (for Machine ID), one or more types, and uses properties from these types in order to provide facts. For example, the Freebase object for Barack Obama has the mid /m/02mjmr and the type /government/us_president (among others)) that allows the entity to have a fact with the property /government/us_president/presidency_number and the literal integer “44” as the value. Freebase uses Compound Value Types to represent n-ary relations with $n > 2$, e.g., values like geographic coordinates, political positions held with a start and an end date, or actors playing a character in a movie. Compound Value Types values are just objects, i.e., they have a mid and can have types [24]. Most non- Compound Value Types objects are called topics in order to discern them from Compound Value Types.

  Google stopped all the Freebase services in 2016 and its data was "donated" to Wikipedia, though only 9.5% of its entities have actually been included in

Wikidata, partly because of the notability criteria mentioned previously. The last dump of Freebase is still available for download[1].

- **Wikidata [23]**
  Wikidata[2], operated by the Wikimedia Foundation is a community-created knowledge base to manage factual information of Wikipedia and its sister projects operated by the Wikimedia Foundation [23]. Wikidata is a collection of entity pages. Entity pages are of two types: items and properties. Every item page contains labels, short descriptions, aliases, statements, and site links. Each statement consists of a claim and one or more optional references. Each claim consists of a property-value pair and optional qualifiers. Values are also divided into three types: no value, unknown value, and custom value. The no value marker means that there is certainly no value for the property, the unknown value marker means that the property has some value, but it is unknown to us, and the "custom value" which provides a known value for the property[2].

Compared with the curated method, the scalability of the collaborative method is far better because of open communities. Although, the collaborative method can create very rich KG, the correctness of KG is still an issue. Any individual can directly manipulate the knowledge graph. Thus, we cannot ensure the quality of the knowledge of the KG. Furthermore, there is no quality measurement in the Wikidata [23]. Moreover, the scalability in the manual approach could be partly solved by the collaborative method. Still, knowledge emerges every day. Therefore, we need an approach that can be done automatically or use less human effort in order to deal with the practical situation in the big data era.

2.3.2 Semi-automatic Approach

In the semi-automatic approach, hand-crafted rules or regular expression rules are manually defined and then such rules are used to automatically extract knowledge from structure data in order to generate triples of a KG. Well-known KG projects, which apply the semi-automatic approach are DBpedia, YAGO, and Freebase. The details of each project is in the following list.

[1] https://developers.google.com/freebase/
[2] https://wikidata.org

Figure 2.3: The sample screenshot of the Wikipedia page about Barack Obama [2]

- **DBpedia [25]**
  DBpedia is the project to construct a KG by extracting knowledge from structure content, specifically Wikipedia. Wikipedia is a free online encyclopedia platform that allows individual users to collaborate with each other for creating web content. In Wikipedia, a user can add update, and remove content on the project directly. Also, one of the advantages of Wikipedia is that a number of hyperlinks among pages are fruitful. Currently, there is much content on Wikipedia since Wikipedia implements the crowdsourcing method to gather the content. In Figure 2.3, the snapshot of the example of the Wikipedia page is presented. As shown in the figure, a Wikipedia page consists of two main parts: 1) description text and 2) infobox. The description text is a text which gives finer details about the pages, while the infobox provides significant information about the page in a well-defined structure format.

  DBpedia, as categorized in the semi-automatic approach, mainly extracts

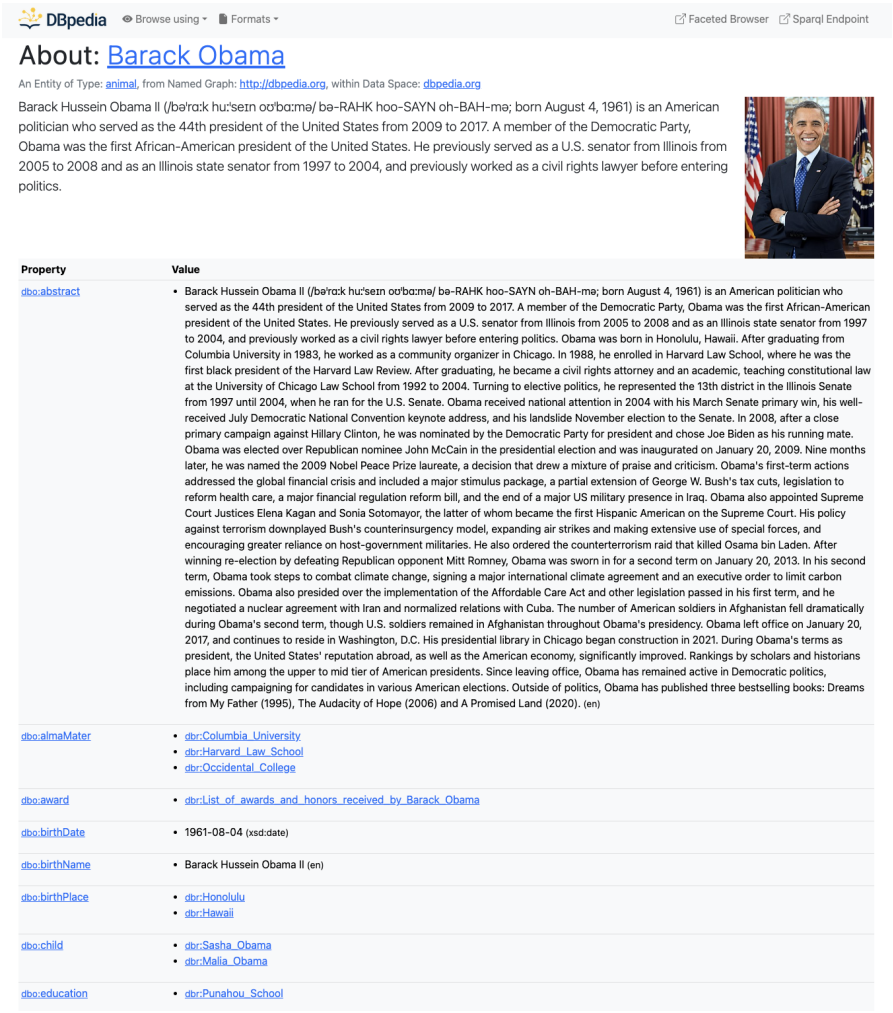

Figure 2.4: The sample screenshot of the DBpedia page dbr:Barack_Obama [3]

knowledge from an infobox and some structured markup of Wikipedia such as abstracts or links. The process of extraction is straightforward. In the process, an entity and a relation are extracted. For an entity, the URI representing the entity of the page is constructed under the namespace of DBpedia (dbr:). For example, considering the example Wikipedia entity “Barack Obama” in Figure 2.3, the URI representation that corresponded to this entity is dbr:Barack Obama. For a relation, properties are extracted under the namespace of DBpedia (dbo: or dbp:) in the same manner as entities from the info box of the Wikipedia page. For example, “Born” in the info box in the example is extracted, and then by applying

the rule, “Born” is mapped to the property dbo:birthDate or dbo:birthPlace of DBpedia. Note that, the difference between dbo: and dbp: is that dbp: is a direct map from the infobox without integrating into DBpedia ontology, while dbo: resolves the integration problem.

Apart from the infobox, the extractor of DBpedia also extracts the content in the HTML markup format such as the abstract of the Wikipedia page, the Wikipedia links to other pages, etc. As a result, the DBpedia entity can be created. In Figure, the snapshot of DBpedia dbr:Barack Obama, which corresponds to the Wikipedia page “Barack Obama” is illustrated in Figure 2.4.

Nowadays, DBpedia becomes one of the most important KGs. Because of the fruitfulness of entities in DBpedia, this KG, therefore, gains more and more attention from many KG publishers as well as researcher communities, e.g. semantic web community, and NLP community. Furthermore, due to the quality of DBpedia and the wide range of languages, which DBpedia provided, it becomes a multilingual KG with high quality [26]. Consequently, many KGs frequently connect their knowledge, including entities and relationships, to DBpedia. As a result, DBpedia becomes the hub of KGs [27].

- **YAGO [28]**
  YAGO is a project similar to DBpedia. It is extract structure knowledge from Wikipedia, e.g. infobox, category, redirected, and Wordnet, e.g. synset, and Geonames [29] in order to create a KG. The project reports that YAGO contains more than 1 million entities and 5 million facts connecting such entities. Furthermore, in the YAGO project, interlinking links between DBpedia and YAGO are provided. Specifically, YAGO provides links to DBpedia ontology [25] and SUMO ontology [30]. In the study [28], the empirical evaluation of the correctness of YAGO is conducted. The result shows that YAGO achieved an accuracy of 95%, which is highly reasonable in the KG construction.

- **Freebase [22]**
  We discussed Freebase in the aspect of manual KG construction in previous Section 2.3.1.2. In fact, Freebase also extracted knowledge from the structure resources such as DBpedia. The idea is similar to DBpedia and YAGO. Freebase

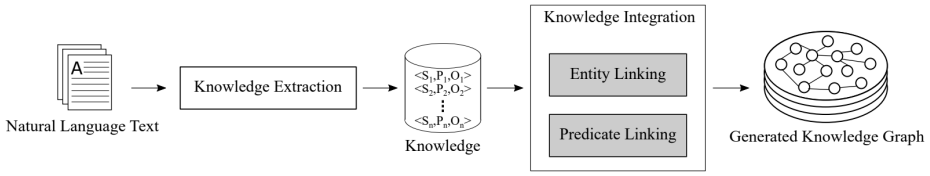

Figure 2.5: The General Work Flow of the Knowledge Graph Population

derived the information in the info box of Wikipedia to create the fact triple. Note that, such create triple are allowed to manually manipulate by a user. Therefore, the Freebase project is a hybrid combination of the semi-automatic approach and the manual approach.

Although the semi-automatic approach can construct effectively, some information or knowledge is still left in the natural language text. For example, considering Figures 2.3 and 2.4, the Wikipedia page described the entity “Barack Obama” contains much more knowledge than DBpedia provided. This characteristic occurs because DBpedia does not directly extract information from the unstructured text, specifically natural language text. Consequently, much knowledge was left as text.

2.3.3 Automatic Approach

In the automatic approach, a KG is directly built from natural language text. An automatic approach extracts knowledge from natural language text and then creates the KG by such knowledge. Generally, The process of the automatic approach could be viewed as shown in Figure 2.5. In the automatic approach, there are two main methods: 1) schema-based method and 2) schemaless-based method. The schema-based method populates knowledge as defined in the predefine ontology or vocabulary. In contrast with the schema-based method, the schemaless-based method extracts the knowledge directly from the natural language text without a predefined vocabulary or an ontology. The details of the schema-based method and the schemaless-based method are as follows.

2.3.3.1 Schema-based Method

The schema-based method aims to build a KG by using a predefine ontology and a finite set of vocabularies as control constraints. In this method, a set of entities and a set of relations are defined; specifically, relations are fixed by a set of predefined vocabularies. When extracting knowledge from the natural language text, this method focuses on extracting the knowledge that corresponds to the given vocabulary. As a result, a set of vocabularies plays a significant role in this method. The details of systems that are categorized into the schema-based method are presented as follows.

- **Never-Ending Language Learning (NELL) [31]**
  NELL is a never-ending learning project, which aims to extract knowledge from the web [31]. The main idea of NELL is to create a set of triples with its ontology by gathering information on the Internet. The main concept of NELL is that NELL will accumulate knowledge over time and due to knowledge acquisition, it becomes better and better to learn new knowledge. In NELL, predefined ontology is defined and some bootstrapping triples together with a set of constraints, including domains and ranges of a relation and mutual-exclusion condition, are given to NELL as the bootstrapping learning data. Such bootstrapping learning data are used to learn constraints for extracting a new belief. One of the prominent features of NELL is that it uses multiple extractors and validators to learn and verify new knowledge. This strategy is called "couple learning". Specifically, NELL uses one extractor to support or argue another extractor in order to populate new knowledge from a set of beliefs. The NELL project started in 2010 and has been continuously running since then. Currently, NELL contains more than 50 million candidate beliefs and more than 3 million beliefs, with high confidence as knowledge.

- **LODifier [32]**
  LODifier is a project to generate a KG from unstructured text. In LODifier, Discourse Representation Structures (DRS) [33] that represent the meaning of a sentence from the unstructured text are extracted by the statistical parser C&C and the semantics construction toolkit Boxer [34]. Also, many NLP systems, including the NER system, Coreference Resolution system, and Word

Sense Disambiguation, are applied and the results are mapped to RDF triples. Furthermore, RDF WordNet [35] is used as the predefine ontology in order to directly map the result, which is a vocabulary from the synset, to an RDF triple without considering other KGs.

- **KnowledgeStore [36]**
  KnowledgeStore is a general-purpose system to extract store and manage knowledge. To achieve the goal the system builds on the state of the art NLP applications, e.g. Tagging system and Coreference Resolution system. The architecture of this system consists of four layers: 1) resource layer, 2) mention layer, 3) entity layer, and 4) context layer. Each layer of the system is designed to deal with resources, mentions, entities, and context. The resource is where an entity is acquired. Mention is the specific object, which is considered in the text, while an entity is a unique object, to which mentions map. This means that different mentions can map to identical entities. The context describes the entity based on a specific context such as time, location, etc. Based on this architecture, the KnowledgeStore can well manipulate and store knowledge; however, integration of knowledge to other KGs still is not taken into account. As a result, the usage of the knowledge is very limited to the local KG, e.g. Trentino [36].

- **RExtractor [37]**
  RExtractor [37] proposes a transformation of unstructured text to an RDF triple by using NLP to extract a triple and then using its own ontology to represent an extracted triple. In their approach, the syntactic structure of a document is exploited by NLP applications and then the predefined ontology in the study [38] is used as the schema when populating knowledge. Even though this approach could extract the entity and the target relation, such an entity still does not integrate into other KGs.

- **Knowledge Vault [39]**
  Knowledge Vault is a project to build a large-scale probabilistic KG based on the combination of content extracted from web documents. In Knowledge Vault, a predefined ontology, including entity type and predicate, is given as

the fixed schema similar to other approaches. The main difference between Knowledge Vault and other systems is that the noise during the extraction process is considered; in consequence, Knowledge Vault becomes more robust. To extract knowledge, several extractors are used to gather knowledge from various types of data, e,g natural language text, tree (DOM), table, etc. Here, we mainly focused on natural language text. To extract knowledge in natural langue text, the following processes are performed. Firstly, entities over all documents are recognized. Secondly, each entity is resolved and linked to KG by using the NLP suit tool [40]. Thirdly, the supervised learning technique, named distant supervision [41], is used to learn the relationship between entities based on the seed triples to find the relationship between entities.

Given the predefine vocabulary, it avoids the heterogeneous problem when populating knowledge; however, the acquired knowledge is very limited due to the condition of predefined vocabulary. To reduce these limitations, the other method, the schemaless-based method, is proposed to extract knowledge. The details of the schemaless-based method are described in the following section.

2.3.3.2 Schemaless-based Method

The schemaless-based method, also known as the Open Information Extraction task [42], aims to build a KG without requiring pre-specified ontology and vocabulary. This method, therefore, has to automatically identify arbitrary relations and extract such relations. In this method, the lexicon plays an important role in the extraction process because the extraction is performed at the lexical level. At the lexical level, there is no require any schema or vocabulary. As a result, the structure of sentences, obtained by a parsing system, significantly helps this method to extract triples. In the following, the systems in the schemaless-based methods are presented.

- **TextRunner [42]**
  TextRunner is a scalable open IE system, which extracts triples from text and assigns a probability to each triple. In the TextRunner system, there are three main components: 1) extractor component 2) Self-Supervised Classifier component, and 3) assessor component. The extractor component, namely the Single-Pass

extractor, reads through the entire document and separately processes each sentence in the document to produce the extraction results; however, this component requires seed triples, as supervision. The Self-Supervised Classifier component module is developed to classify whether extracted triples are trustable or not and then trustable triples are given back to the extractor component as seed triples. The assessor component validates and judges the probability score for each extraction result.

- **ReVerb [43]**
  ReVerb is an open information extraction system that extracts a triple from a given sentence by using syntactic patterns and lexical constraints. In ReVerb, a relation phrase is identified by using syntactic patterns and lexical constraints. For the syntactic patterns, prior knowledge regarding the language is provided such as "phrase relation must start with a verb and end with the preposition". Such syntactic patterns are used to identify useful syntactic to extract triples. On the other hand, the lexical constraints help to generalize the extracted results. For example, some relation between entities might be extracted with the long phrase of relation, meaning that it is too specific. To avoid such problems the lexical constraints are applied. Then, entities, and noun phrases that correspond to related phrases are assigned. Finally, the confidence score for the extraction triple is given and adjusted.

- **OLLIE [44]**
  OLLIE system is an open information extraction system, which has been improved from ReVerb [43]. The OLLIE system works in a similar manner to ReVerb; however, the OLLIE system can extract finer details in the local context of a sentence. ReVerb constraints mainly focused on the main verb of the sentence. As a result, some latent relation is missing. In OLLIE, patterns are not limited to a main verb of a sentence or a local context of a sentence. For example, "The President of the United State Donald Trump announces the new regulation". ReVerb focuses only on the relation "announce", while OLLIE can extract the "President of United States" relation.

Table 2.2: The Summary of the Knowledge Graph Construction projects

Approaches	Methods	Projects
Manual	Curated	Cyc/OpenCyc, WordNet, UMLS
	Collaborative	Freebase, Wikidata
Semi-Automatic		DBpedia, YAGO, Freebase
Automatic	Schema-based	NELL, Exner et al, KnowledgeStore, Kriz et al, Knowledge Vault, LODifier
	Schemeless-based	TextRunner, ReVerb, OLLIE, Exner et al

- **Entity Extraction [45]**
  Entity Extraction proposed the pipeline system to take natural language, specifically Wikipedia articles as input and yielded the KG triple as the output. The idea of the system is to use state-of-the-art natural language processing tools, e.g. a semantic role labeler (SRL) and name entity resolution, and so on, to extract the relation from the text. Then, the system links extracted entities to KG entities and determines the statistical pattern of the text predicate and the KG predicate based on each subject-object pair, and then forms a link between identical predicates. Since the study used an SRL tool to analyze the relation without the predefined vocabulary. Therefore, the system has been categorized into the schemaless-based method. Nevertheless, knowledge integration with the predefined vocabulary is applied as well. Therefore, this approach could also be viewed as the schema-based method.

The schema-based method helps us to populate knowledge for the existing KG; however, we can populate some of the knowledge from text. In the schema-less-based method, lexicon term plays an important role in the knowledge extraction process and it is not dependent on any vocabulary; in consequence, the schema-less-based method can populate more knowledge but there are not much useful because of the heterogeneous problem. In this research, we aim to take advantage of the schema-based and schema-less-based methods in order to build the KG with wide coverage.

In this section, we reviewed and surveyed various approaches and methods in each approach for the KG construction task. To sum up our discussion so far, we present the summary of the approaches, the methods, and their corresponding projects in Table 2.2.

2.4 Knowledge Graph Completion

KG completion is a task to fill a missing knowledge in KG. As we know KG is incomplete, KG completion uses the current structure or knowledge in KG to find whether there are any other missing relations in KG or not. This task is also widely known as Link Prediction [13] since the scenario in the KG completion is that the missing linkings between entities are predicted. In the KG completion task, there are many approaches [46, 47, 48, 49, 50, 51] proposed so far. A traditional approach for KG completion is to use the association rule to mine rules for filling the knowledge. In contrast with the traditional approach, a modern approach uses the embedding method to embed an entity in the KG so that the links between entities can be predicted. In the following list, AMIE [46], which is a traditional approach, and TransE [48], which is a modern approach, are presented since they are fundamental to these approaches.

- **AMIE [46]**
  AMIE is a rule-mining system, which extracts logical rules, specifically Horn clauses [52]. The AMIE system is designed to work on Open World Assumptions. If the logical rules do not contradict with association rules that discover by the system, a triple cannot be identified whether it is correct or not. In order to create rules, an efficient association rule mining algorithm is proposed to deal with the large scale of KG. One major contribution is to simulate negative triples in KG. Since the association rule mining algorithm requires negative samples when learning the rule, Closed World Assumption is applied in the association rule learning state. An example of a learning rule is "isDirectedBY(movie, person)" [46]. After acquiring the rules such rules are used to populate a new triple which not exist in the current KG.

- **TransE [48]**
  TransE is a pioneer research project for the KG embedding task. The KG embedding task is to represent each element of triples in KG into the continuous vector space similar to the word representation [53]. After embedding elements in KG in the distributed representations, such representation can be used to predict the missing relation in the KGs. This goal can be accomplished because the objective function of KG embedding is to try to minimize the error caused by

a particular relation and two entities and maximize the non-existing relation. More Formally, given a triple (h, l, t), the main assumption of the KG embedding is that $h + l \approx r$. Therefore, the objective function of transE is to optimize Equation 2.1.

$$minimize_{h,r,l,h',l',r'} |d(h + l, r) - d(h^{'} + l^{'}, r^{'})| \tag{2.1}$$

where (h, l, r) is an existing triple, $(h^{'}, l^{'}, r^{'})$ is a non-existing triple, d(·) is a distant metric, e.g., Euclidean distance. Based upon the inspiration of the TransE method, many studies further investigate the KG embedding method to tackle the KG completion task. As a result, the variation of the TransX models. e.g., TransH [49], TransR [50] and TransG [51], are proposed. where (h, l, r) is an existing triple, (h0, l0, r0) is a non-existing triple, d(·) is a distant metric, e.g., Euclidean distance. Based upon the inspiration of the TransE method, many studies further investigate the KG embedding method to tackle the KG completion task. As a result, the variation of the TransX models. e.g., TransH [49], TransR [50] and TransG [51], are proposed.

As discussed above, KG completion is to predict the missing relation between entities in a KG. Techniques used in this task are totally different from the KG construction. One prominent aspect is that the KG completion does not get involved with other natural language texts when completing knowledge in a KG. This implies that the external knowledge resources have not been used. Furthermore, KG completion approaches simply find the missing link; however, a non-existing entity is out of the scope of this research topic.

2.5 Natural Language Processing

2.5.1 Vector Representation of Words

Machines are better at understanding numbers than actual text passed on as tokens. This process of converting text to numbers is called vectorization. Vectors then combine to form vector space which is continuous in nature, an algebraic model

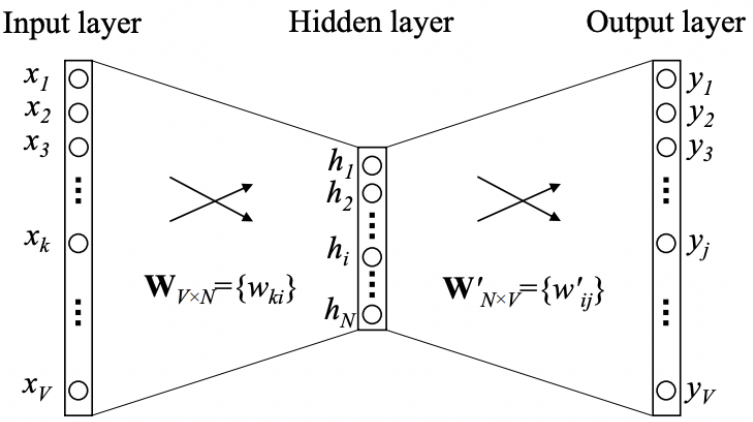

Figure 2.6: A simple CBOW [4] model with only one word in the context

where rules of vector addition and similarity measures apply. Different approaches to vectorization exist.

2.5.1.1 Word2vec

Word embedding actually refers to the numerical representation of words. We commonly use similar colors in the form of RGB. Word2Vec basically means expressing each word in your text corpus in an N-dimensional space often referred to as embedding space. The simplest word embedding can be done by using one-hot vectors. If the word corpus contains 10,000 words as vocabulary, then one-hot encoding can represent each word as a 1x10,000 vector. The reason for choosing one-hot encoding is due to simplicity, robustness, and observation that simple models trained on huge amounts of data outperform complex systems trained on fewer data [54]. The Word2vec model captures both syntactic and semantic similarities between the words[3]. One of the well-known examples of the vector algebraic on the trained word2vec vectors is Vector("France")-Vector("Pais")= Vector("Tokyo")-Vector("Japan").

Word2Vec is a predictive embedding model. Predictive models learn their vectors

[3] https://towardsdatascience.com/word2vec-research-paper-explained-205cb7eecc30

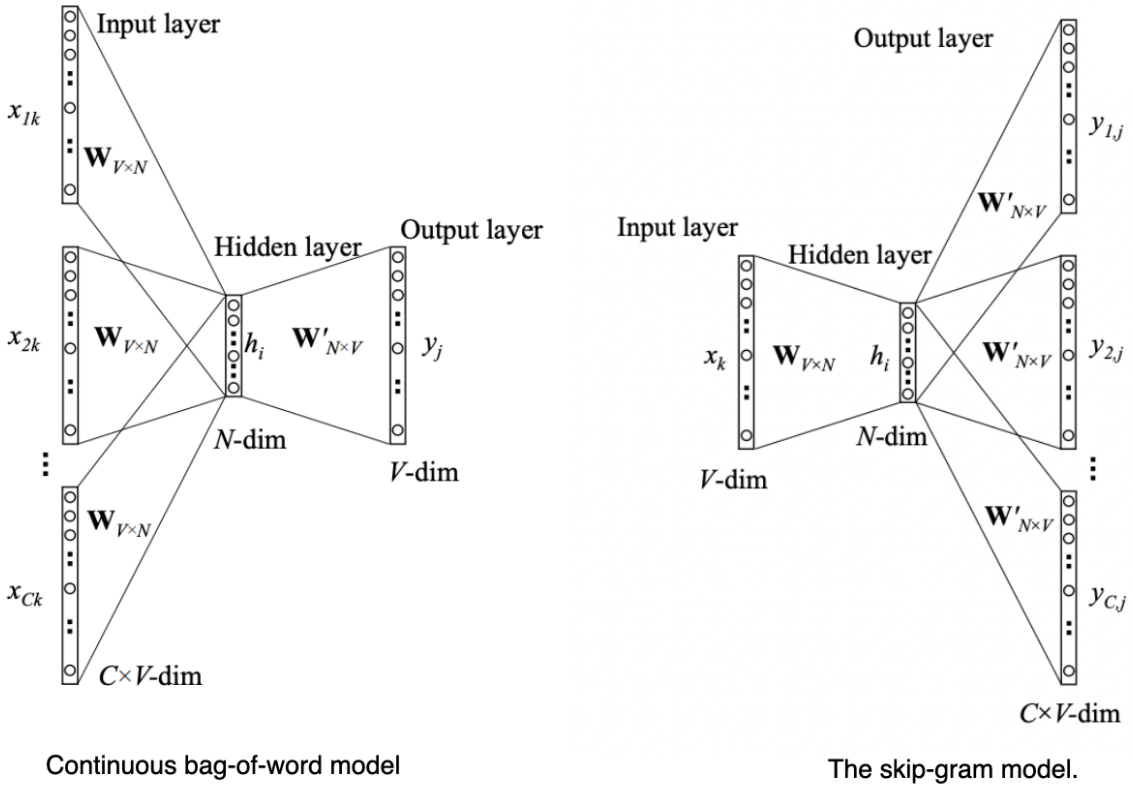

Figure 2.7: Continuous bag-of-word model and Skip-gram Model [4]

in order to improve their predictive ability of a loss such as the loss of predicting the vector for a target word from the vectors of the surrounding context words. There are two main Word2Vec architectures that are used to produce a distributed representation of words: 1) Continuous Bag of Words (CBOW) and Skip-gram. We will discuss them in detail in the following sections.

2.5.1.2 Continous Bag of Words (CBOW)

In the continuous bag of words (CBOW) model, the distributed representations of context (or surrounding words) are combined to predict the word in the middle. When there is only one word per context, the model will predict one target word given one context word. Figure 2.6 shows the network model with only one word in the context.

Here, the vocabulary size is V, and the hidden layer size is N. The units on adjacent layers are fully connected. The input is a one-hot encoded vector, which means for a given input context word, only one out of V units, x1, , x_V , will be 1, and all other units are 0. Figure 2.7 shows the CBOW model with a multi-word context setting. When computing the hidden layer output, instead of directly copying the input vector of the input context word, the CBOW model takes the average of the vectors of the input context words, and use the product of the input→hidden weight matrix and the average vector as the output [4]. Total weights involved in training CBOW model are N×D+D×log(2)V.

2.5.1.3 Skip-gram

Skip-gram model, the distributed representation of the input word is used to predict the context. Figure 2.7 shows the Skip-gram model. It is the opposite of the CBOW model. The target word is now at the input layer, and the context words are on the output layer. Skip-gram works well with a small amount of the training data and represents well even rare words or phrases. CBOW is several times faster to train than the skip-gram, with slightly better accuracy for the frequent words. The total complexity of the model is N×D+N×D×log2(V). Noticeably, N also gets multiplied by D×log2(V) term as it's not a single class classification problem compared to CBOW, but rather N class classification problem. Hence overall complexity of skip-gram model is greater than the CBOW model[3].

2.5.2 Recurrent Neural Networks

Recurrent Neural Networks (RNN) have shown promising results in processing arbitrary sequences of input. For a given sequence of input $x_1, x_2, ..., x_n$, the RNN model learns the current latent state with the input data at time t and the previous latent state at time $t-1$. Then the current latent state is used to predict the output. The RNN is derived as follows:

$$h_t = f(W_{i,h}x_t + W_{h,h}h_{t-1} + b_h)$$
$$y_t = g(W_{h,y}h_t + b_y)$$

where x_t is the input vector at time t, h_t is the vector of hidden layer at time t, y_t is the prediction vector at time t, $W_{i,h}$, $W_{h,h}$, $W_{h,y}$ are parameter matrices, b_h, b_y are the bias parameters for the network and f, g are the activation functions, e.g., sigmoids.

Although RNN is able to handle a variable-length sequence input, long-term dependencies are difficult to be captured due to the gradients that tend to either vanish or explode. The long short-term memory (LSTM) unit and gated recurrent unit (GRU) are able to handle long-term dependencies and perform better than using traditional *tanh* unit [55].

2.5.3 Gated Recurrent Unit

A gated recurrent unit (GRU) was proposed by Cho et al. to make each recurrent unit to adaptively capture dependencies of different time scales [55]. Similar to the LSTM unit, the GRU has gating units that modulate the flow of information inside the unit, however, without having a separate memory cells. The GRU is like a long short-term memory (LSTM) [56] with a forget gate [57], GRU has fewer parameters than LSTM, as it doesn't have an output gate. GRU's performance on specific tasks of polyphonic music modeling, speech signal modeling and natural language processing is similar to LSTM models, but it has shown better performance on certain smaller and less frequent datasets.

There are several variations on the full gated unit, with gating done using the previous hidden state and the bias in various combinations, and a simplified form called minimal gated unit.

A fully gated unit is defined as follows:

$$
\begin{aligned}
z_t &= f(W_z x_t + U_z h_{t-1} + b_z) \\
r_t &= f(W_r x_t + U_r h_{t-1} + b_r) \\
h_t &= g(W_h x_t + U_h(r_t \circ s_{t-1}) + b_h) \\
s_t &= z_t \circ s_{t-1} + (1 - z_t) \circ h_t
\end{aligned}
$$

where x_t is the input vector at time t, z_t is the update gate vector at time t, r_t is the reset gate vector at time t, h_t is the hidden layer vector at time t, s_t is the output vector at time t, W and U are parameter matrices, b is bias parameter, f and g are

activation functions, and ∘ is the Hadamard product operation. σ_g is the sigmoid and ϕ_h s a hyperbolic tangent activation function.

2.5.4 BERT

BERT (Bidirectional Encoder Representations from Transformers) [5] is a transformer-based machine learning technique for natural language processing pre-training developed by researchers at Google AI Language. When BERT was published, it achieved state-of-the-art performance on a wide variety of NLP tasks, including GLUE (General Language Understanding Evaluation) task, and Question Answering (SQuAD v1.1). BERT achieved remarkable performance in Natural Language Inference (MNLI), SWAG (Situations With Adversarial Generations), Sentiment Analysis, and others.

BERT's key innovation is applying the bidirectional training of the Transformer. For language modeling, Transformer is a popular attention model. But previously, text sequences were looked at either from left to right or combined with left-to-right and right-to-left training. BERT shows that if the language model is bidirectionally trained, it can have a more profound sense of language context and flow, which is not possible in single-direction language models. BERT framework consists of two basic steps: pre-training and fine-tuning.

2.5.4.1 BERT Pre-training

During pre-training (see, Fig 2.8), the model is trained on unlabelled data over different pre-training tasks. BERT uses two unsupervised tasks for pre-training named as: Masked LM and Next Sentence Prediction (NSP).

Masked LM (MLM) For training, 15% of the words in each sequence is replaced with a [MASK] token before feeding into BERT. The model then tries to predict the original value of the masked words based on the context provided by the other words in the sequence.

Technically, the prediction of the output words requires [4] :

- Adding a classification layer on top of the encoder output.

[4]https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

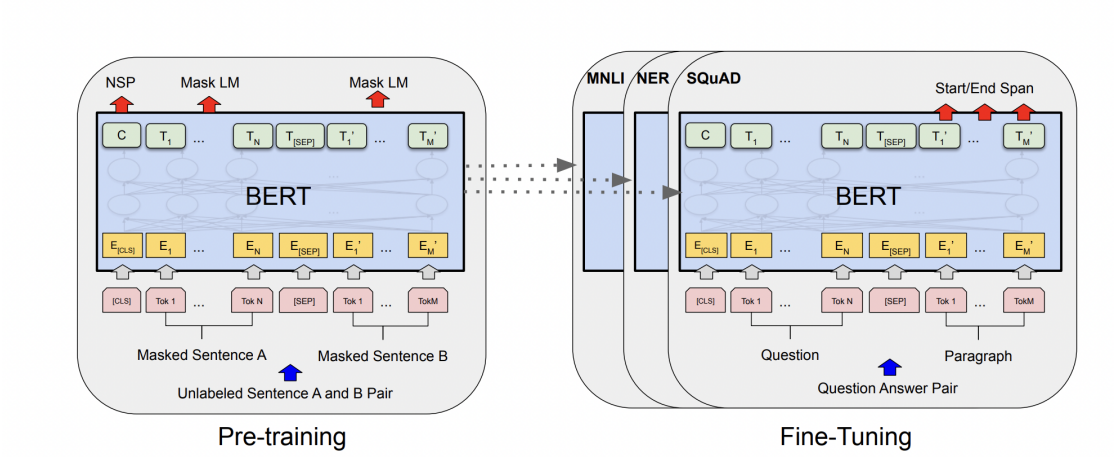

Figure 2.8: Overall pre-training and fine-tuning procedures for BERT [5]

- Multiplying the output vectors by the embedding matrix, transforming them into the vocabulary dimension.

- Calculating the probability of each word in the vocabulary with softmax.

Next Sentence Prediction (NSP)

Many important downstream NLP tasks are based on understanding the relationship between two sentences. This is not directly captured by previous language modeling. In understanding sentence relationships, BERT pre-trains for a binarized next sentence prediction (NSP) task that can be trivially generated from any monolingual corpus. Specifically, when choosing the sentences A and B for each pretraining example, 50% of the time B is the actual next sentence that follows A (labeled as IsNext), and 50% of the time it is a random sentence from the corpus (labeled as NotNext) [5]. A [CLS] token is inserted at the beginning of the first sentence and a [SEP] token is inserted at the end of each sentence. A sentence embedding indicating Sentence A or Sentence B is added to each token. Sentence embeddings are similar in concept to token embeddings with a vocabulary of 2. A positional embedding is added to each token to indicate its position in the sequence[4].

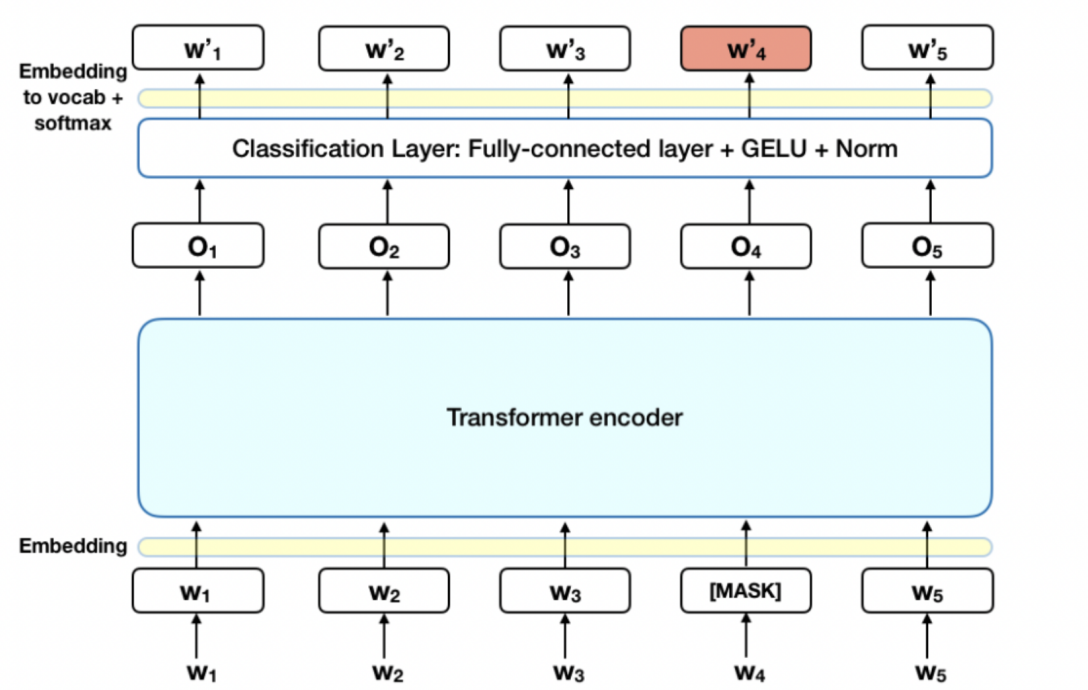

Figure 2.9: MaskLM

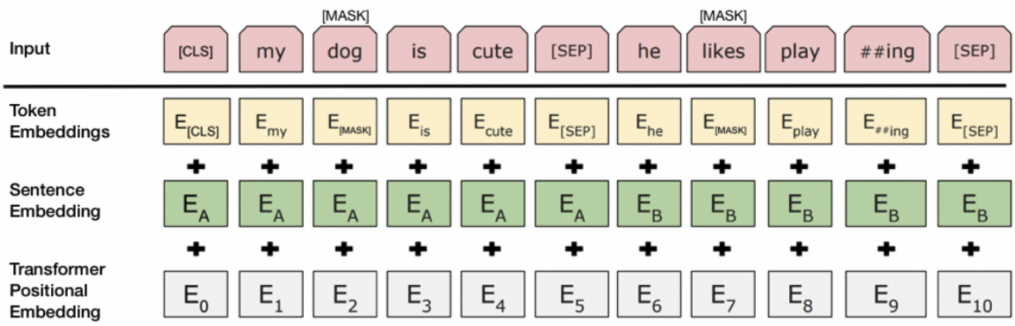

Figure 2.10: Next Sentence Prediction [5]

2.5.4.2 BERT Fine-training

Fine-tuning is straightforward since the self-attention mechanism in the Transformer allows BERT to model many downstream tasks. Classification tasks such as sentiment analysis are done similarly to Next Sentence classification, by adding a classification layer on top of the Transformer output for the [CLS] token. In Question Answering tasks, the software receives a question regarding a text sequence and is required to mark the answer in the sequence. Using BERT, a Q&A model can be trained by learning two extra vectors that mark the beginning and the end of the answer[4]. Compared to pre-training, fine-tuning is relatively inexpensive.

2.6 Open Information Extraction

Open Information Extraction (OIE) systems aim to extract unseen relations and their arguments from unstructured text in an unsupervised manner. In its simplest form, given a natural language sentence, they extract information in the form of a triple, consisting of subject (S), relation (R), and object (O).

Suppose we have the following input sentence: *AMD, which is based in U.S., is a technology company.* An OIE system aims to make the following extractions: *("AMD"; "is based in"; "U.S.") ("AMD"; "is"; "technology company")*

Most commonly, OpenIE systems extract schemaless triples from an input sentence. In principle, OpenIE representations represent knowledge that is found in natural language sentences into structured machine-readable form. Contrary to traditional information extraction pipelines, OpenIE systems do not require predefined schemas. Standard IE systems are limited by predefined schemas, which makes them unable to extract information that goes beyond the schemas. On the other hand, OpenIE systems, in principle, are able to extract any form of relation between two entities, which makes them scalable w.r.t. the diversity of natural language.

OpenIE extractions are useful for numerous downstream tasks, including question answering [58, 59, 60], information retrieval [61, 62], slot filling [63, 64], event schema induction [65], text summarization [66], knowledge base population [67, 68], entity aspect linking [69], link prediction [70] and open link prediction [71].

2.7 Close-world assumption vs Open-world assumption

The Closed World Assumption (CWA) is the assumption that what is not known to be true must be false. The Open World Assumption (OWA) is the opposite. In other words, it is the assumption that what is not known to be true is simply unknown.

Consider the following statement: *"Juan is a citizen of the USA."* Now, what if we were to ask *"Is Juan a citizen of Colombia?"* Under a CWA, the answer is no. Under the OWA, it is I don't know.

2.7.1 When do CWA and OWA apply?

The CWA applies when a system has complete information. This is the case for many database applications. For example, consider a database application for airline reservations. If you are looking for a direct flight between Austin and Madrid, and it doesn't exist in the database, then the result is *"There is no direct flight between Austin and Madrid."* For this type of application, this is the expected and correct answer.

On the other hand, OWA applies when a system has incomplete information. This is the case when we want to represent knowledge (a.k.a Ontologies) and want to discover new information. For example, consider a patient's clinical history system. If the patient's clinical history does not include a particular allergy, it would be incorrect to state that the patient does not suffer from that allergy. It is unknown if the patient suffers from that allergy unless more information is given to disprove the assumption.

2.7.2 CWA vs OWA: an example

CWA is not only about returning "no" and OWA is not only about returning "I don't know." Consider the following example:

Let's continue with the example of "Juan is a citizen of USA" and assume the following statement is true: "a person can only be citizen of one country." Up to now, everything is fine. Now consider we add the following statement: "Juan is a citizen of Colombia." In a CWA system, this would be an error because we previously stated that

person can only be a citizen of one country and we assume that USA and Colombia are different countries. In an OWA system, instead of generating an error, it would infer a new statement. The logic is the following: "If a person can only be citizen of one country, and if Juan is a citizen of USA and Colombia, then USA and Colombia must be the same thing!"

Note that in the CWA case, we assumed that USA and Colombia are different countries. With OWA, this is not assumed. This is what is called Unique Named Assumption (UNA). CWA systems have UNA. OWA systems do not have UNA. However, one could manually add the UNA. In other words, if I have a list of all the countries, I would have to explicitly state that each country is different from each other. In our example, if we add the following statement: "USA is different from Colombia," the OWA would now generate an inconsistency. The OWA logic is the following: "If a person can only be a citizen of one country, and if Juan is a citizen of USA and Colombia, then USA and Colombia must be the same thing; but hold on, USA and Colombia are different, so they can't be the same! Something is wrong."

2.7.3 OWA and the Semantic Web

Recall that OWA is applied in a system that has incomplete information. Guess what the Web is? The Web is a system with incomplete information. Absence of information on the web means that the information has not been made explicit. That is why the Semantic Web uses the OWA. The essence of the Semantic Web is the possibility to infer new information.

2.8 OpenKG

Existing KGs are mainly constructed under the closed-world assumption and those are in a fixed structure-based format. Therefore, adding new information is very complex. On the other hand, under the open-world assumption, all entities and relations are not known previously and we can utilize natural text for adding new information. Here, by utilizing OpenIE tools we can extract triple-format information from text those can utilize to construct OpenKG.

3 Competent Triple Identification

3.1 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 [72], YAGO [73], and Freebase [74]. 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 [6] and ComplEx [7] 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 [8]. 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) [9, 10, 11] 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.

3.2 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 τ 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,

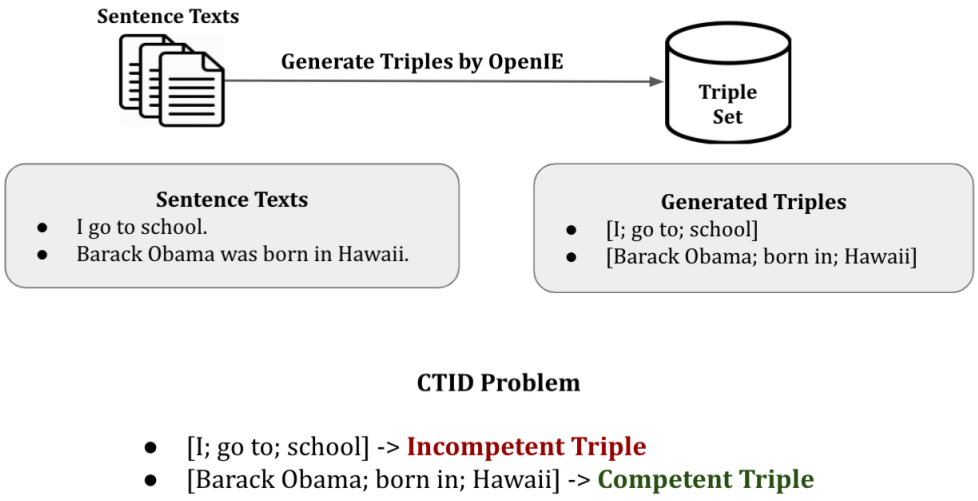

Figure 3.1: Illustration of CTID problem. Triples generated by OpenIE can be noisy. The CTID model can effectively identify competent triples for KGs.

$\exists O_e \notin E$ 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 τ, we identify *competent* triples, which can be used for KG completion. An illustration of this problem is shown in Figure 3.1. 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.

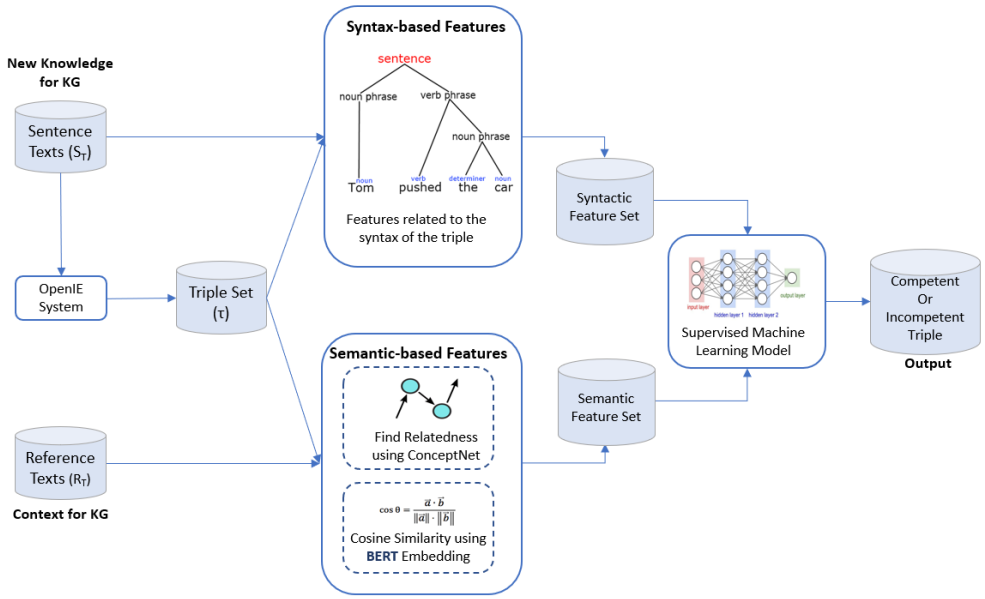

Figure 3.2: Overview of CTID model for identifying competent triples from unstructured text. For the extracted triple set τ, the proposed syntax- and semantic-based features are prepared separately. Then, a supervised model is applied to classify the triples.

3.3 Related Work

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 [6, 75, 76, 77] 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 [78] and GRank [79], 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 [80, 81] 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 [82]

and OLLIE [83] 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 [84] 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 [85, 86] 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 [87], 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.

3.4 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 [83]. 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.

Table 3.1: 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)

3.4.1 ReVerb System

In our approach, we utilize the syntactic and lexical constraint mechanisms of the ReVerb system [82]. 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 [88] and WOE [89], 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 3.1. 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 3.2 to assign a confidence score to each extracted triple. The features have weights in the confidence calculation.

3.4.2 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 Figure 3.2. 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

Table 3.2: 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 3.1
0.39	The last preposition in r is to
0.25	The last preposition in r is in
0.23	10 words $< len(s) \leq 20$ 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 words $< len(s)$
-0.61	r matches V from Table 3.1
-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

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* τ for each *reference text* r_t. To identify *competent* triples from the *triple set* τ, 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

Table 3.3: 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

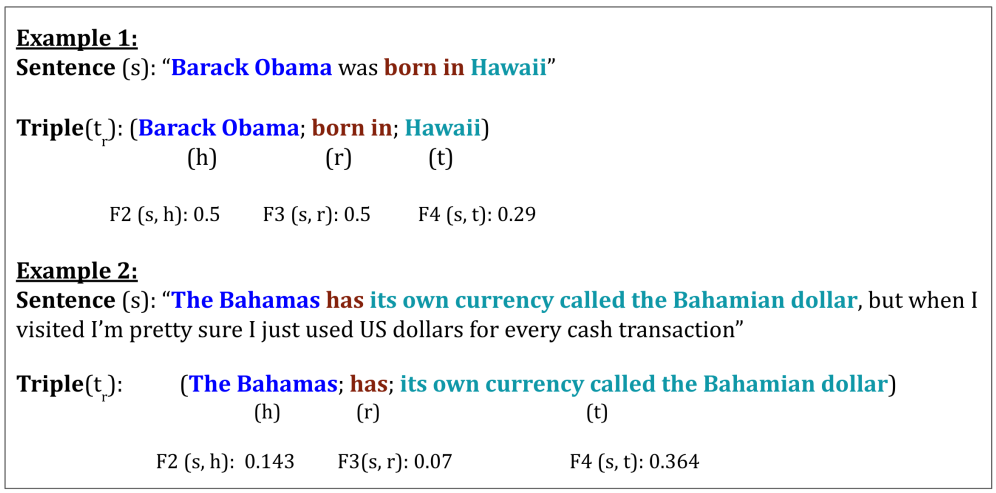

Figure 3.3: Description of proposed features F2, F3, and F4. Please refer to Section 4.2.1 for details.

is used to classify a triple as *competent* or *incompetent*. The proposed features are described in detail below.

3.4.2.1 Syntax-based Features

Syntax-based features mainly deal with the syntax of each triple. Here, we propose four syntax-based features (see Table 3.3). 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 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.

$$Dice\_coefficient(X, Y) = \frac{2|X \cap Y|}{|X| + |Y|} \tag{3.1}$$

Figure 3.3 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 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 3.2). 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 3.2 and four proposed features.

3.4.2.2 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) 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*[1] (commonly used to compute semantic similarity) and a cosine similarity that uses BERT embedding [5] (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*

[1]http://conceptnet.io/

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[2], 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.

$$Semantic\_Relatedness\_Measure = W_R \cap W_T \tag{3.2}$$

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

Figure 3.4 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 Figure 3.4, we get common words for $Rel(money)$, $Rel(dollar)$, and $Rel(currency)$. They represent the semantic relatedness among each other. This example shows that

[2]https://spacy.io/

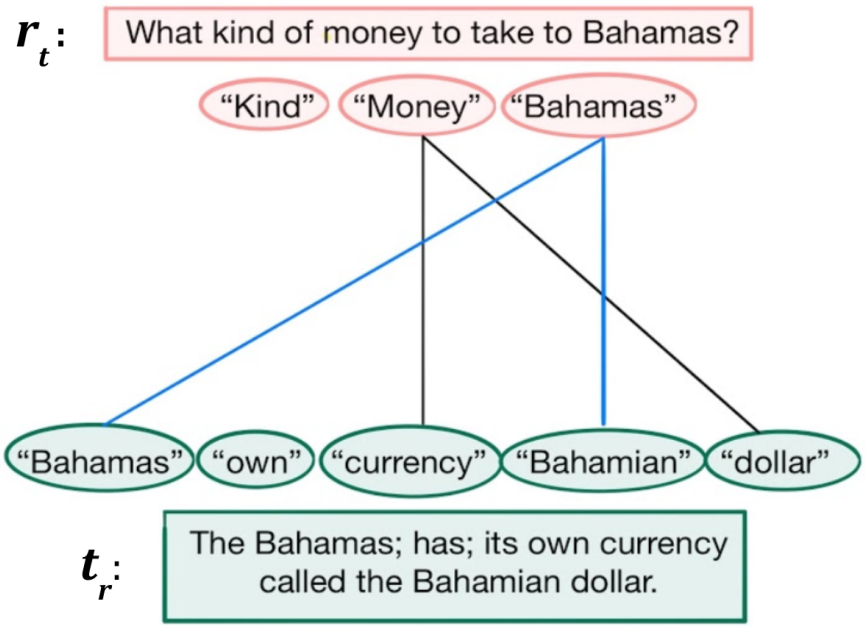

Figure 3.4: Token-based Semantic relatedness measure based on ConceptNet. "Money", "currency", and "dollar" are semantically related here.

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 [5] embedding to find a word vector for each token. Other popular word embedding models such as skip-gram [54], CBOW [54], and GLOVE [90] 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 [5] 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.

$$Similarity\_Measure = \sum_{i=1}^{a} \sum_{j=1}^{b} f(x_i, y_j) \tag{3.3}$$

where,

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

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

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 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.

3.4.2.3 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 3.2 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*.

3.5 Experiment 1

For the evaluation of our CTID model, we conducted two experiments. In the first experiment, we built our dataset and annotated the dataset using an algorithm with some existing information. Next, we identified some limitations of the annotation

algorithm and applied ground truth labeling. In our second experiment, we conducted our experiment which used this ground truth labeling dataset. In this section, we explain our first experiment.

3.5.1 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 [91]. In this dataset, questions are generated based on Freebase [74], 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.

3.5.1.1 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 Figure 3.5. 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 Figure 3.5. 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 [9] system for generating triples. We used OpenIE v4, which is a combination of SRLIE [92] and RelNoun [93]. Table 3.4 shows an example of triple generation using OpenIE v4.

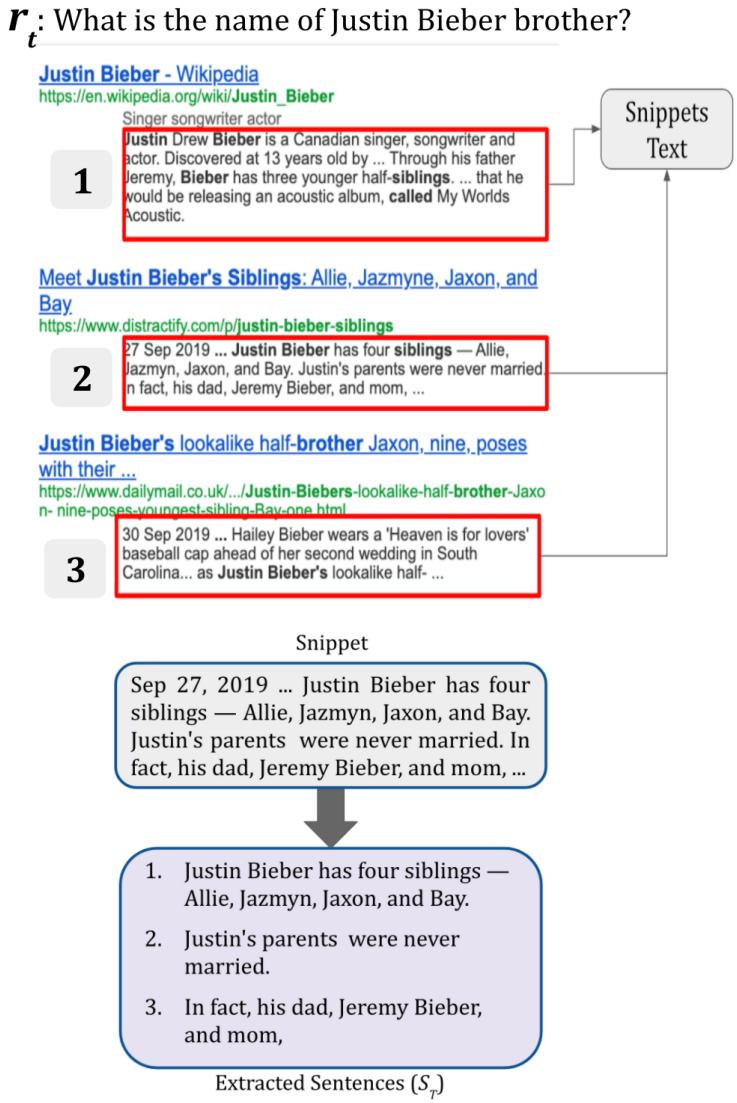

Figure 3.5: Sentence extracted from Google search results. For each reference text, the top 10 relevant snippets are first extracted. The sentences are then separated using text processing.

3.5.1.2 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,

Table 3.4: 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)

rather than three. Hence, we also need to remove incomplete triples from the extracted triple set τ.

3.5.1.3 Data Annotation

We divided all triples into two classes, namely *incompetent* and *competent*. As mentioned in Section 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. To annotate the extracted triples, we utilized the answer entity. We propose a procedure for automatically annotating extracted triples.

Algorithm 3.1 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 3.6 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).

Algorithm 3.1 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

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 the other two triples as incompetent. In Figure 3.6, green indicates a competent triple, and red indicates an incompetent triple.

3.5.2 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

Table 3.5: 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 using Algorithm 3.1	1143

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 Section 5.2.1. Table 3.5 shows a summary of the dataset used in this experiment. We named this dataset "QA2TEXT". There are 1143 competent triples in total. For the validation, we manually check the triples after annotation.

3.5.2.1 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

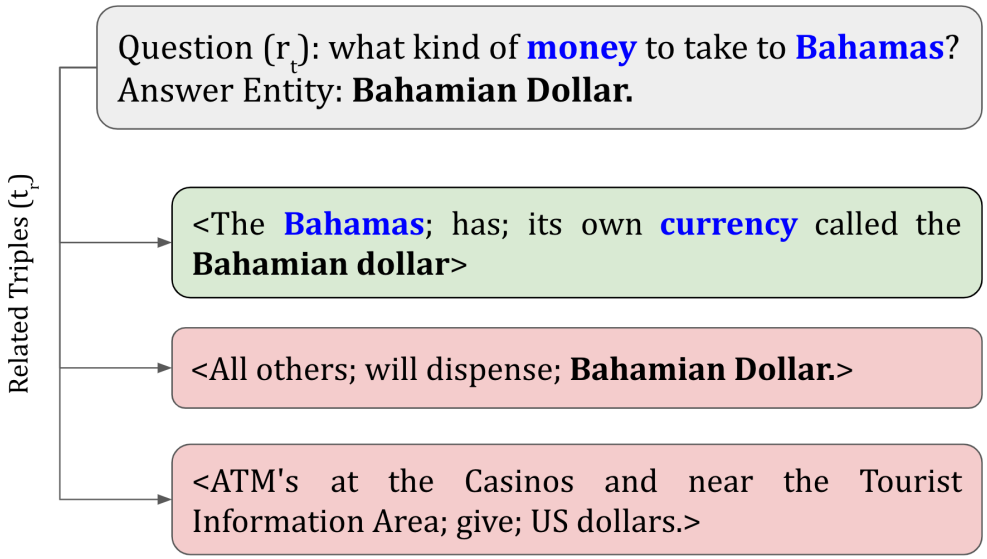

Figure 3.6: Example of data annotation. Green and red boxes respectively represent competent and incompetent triples.

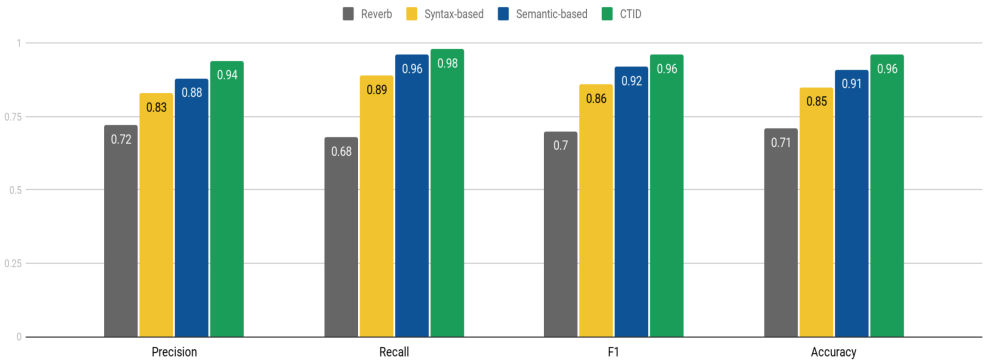

Figure 3.7: Experimental Result. Comparing the performance of proposed features.

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 the sigmoid activation function.

3.5.2.2 Evaluation Measures

For the evaluation, the standard information extraction measures (i.e., precision, recall, F1 score, accuracy) were applied. To explain these evaluation measures we need to define some terms. Those are given below:

C = Identified triples as competent by CTID model those are also annotated as competent
T = Identified triples as competent by CTID model
TC = Total number of annotated competent triples
CI = Correctly identified triples by CTID model
N = Total number of triples

Using the defined terms, the evaluation measures are defined 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.

$$P = \frac{\mathrm{C}}{\mathrm{T}} \tag{3.4}$$

- *Recall:* Recall R represents the degree of correct information retrieved. Therefore, it is the proportion of competent triples that the system assigns to this class. It is calculated using Eq. 5.

$$R = \frac{\mathrm{C}}{\mathrm{TC}} \tag{3.5}$$

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

$$F1 = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.6}$$

- *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.

$$A = \frac{\mathrm{CI}}{\mathrm{N}} \tag{3.7}$$

3.5.2.3 Baseline Model

To design our baseline model, we utilized the features in the ReVerb system[82] (see Section 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.

3.5.3 Experimental Results

Figure 3.7 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 3.7 shows that with the combination of the proposed features and the ReVerb features, our model

Table 3.6: 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

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 3.7 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 in 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 features 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.

3.6 Experiment 2

In our first experiment, we used automated annotation and found some limitations of this annotation procedure. Therefore, we conducted our second experiment. Here, we use human intervention for ground truth labeling. In this section, we explain our second experiment.

3.6.1 Limitations of Annotation Algorithm

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 3.6 shows some of the input triples and the model output with correct answers.

The output shows that triples, which are 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 utilize the automated annotation procedure to assist human-level annotation.

3.6.2 Ground-truth Labeling

For our first experiment, we build up an algorithm for data annotation, in which we use the answer entity available in the WebQuestionsSP dataset. As this algorithm does not assure the ground truth of the annotated triples, we did a further analysis using ground truth labeling. Here, we annotate triple manually with human supervision. Table 3.7 shows the summary of ground truth labeling. Here, the total number of competent triples identified by our annotated algorithm is decreased after adding human label annotation. Some triples are not correctly identified by the algorithm.

Here, we also investigate what types of triples are not identified using our annotated algorithm. As we utilize the WebQuestionsSP dataset for creating our dataset, some

Table 3.7: Summary of Ground Truth labeling

Total Number of Triples	61500
Annotated by our Algorithm	
Number of Competent Triples	1143
Number of Incompetent triples	60357
Result of Ground Truth labeling	
Number of Competent Triples	1089
Number of Incompetent Triples	60411
Incompetent but identified as Competent by our Algorithm	395
Competent but identified as Incompetent by our Algorithm	339

information is changed over time which is identified as "Incompetent" by our annotation algorithm. In Table 3.8, the first example shows this fact. Where the question is "Who is the prime minister of Ethiopia now?" and the given answer entity is "Hailemariam Desalegn" who was the prime minister of Ethiopia before 2018. In 2018, the new prime minister is "Abiy Ahmed". As annotated algorithm uses the given answer entity, here it could not match the answer in the triple part. That's why it is identified as an "Incompetent" triple. Another type of fact is shown in Table 3.8 example 2. Here. the given answer entity "Palestrina" is present in the question. But they are different in context. In the question, "Palestrina" is part of a person's name whereas in the answer entity, "Palestrina" is the name of a place. Our annotation algorithm could not distinguish the same entity name with different meanings. Therefore, using this algorithm these types of triples are identified as "Competent" triples although these

Table 3.8: Examples of facts for not identifying triples by our annotation algorithm

Example	**Fact**
1. Question: Who is the prime minister of Ethiopia now? Answer Entity: Hailemariam Desalegn **(before 2018)** Triple: Abiy Ahmed; [is]; prime minister [of] Ethiopia **(current)**	Information is changed over time
2. Question: Where did Giovanni Pierluigi da **Palestrina** live? Answer Entity: **Palestrina** Triple: Giovanni Pierluigi da **Palestrina;** [is]; Italian Renaissance composer of more than	Same entity different meaning

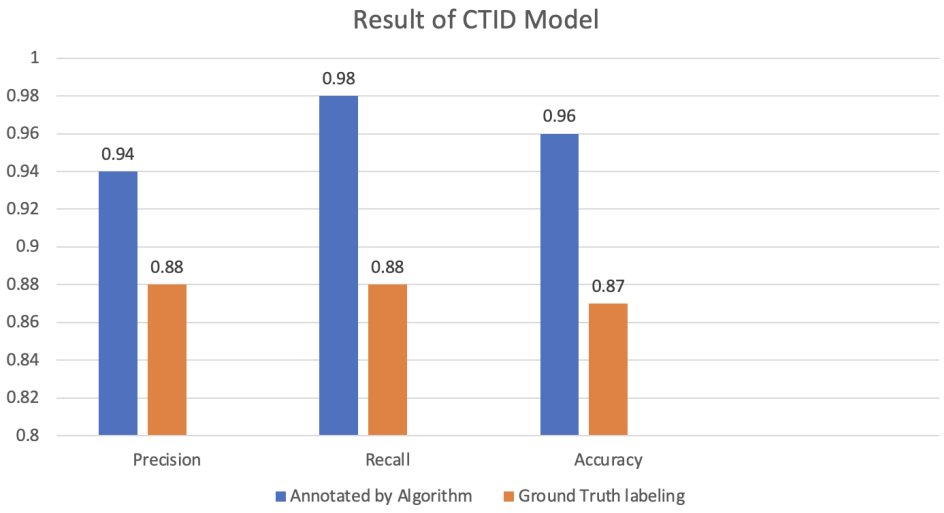

Figure 3.8: Experimental Result. Comparing the performance using ground truth labeling

triples are “Incompetent”.

Using human label annotation, we can remove this type of wrong identification. Here, we check all the triples and correct the labeling done by our annotation algorithm if necessary.

3.6.3 Experiment using Ground-truth Labeling

Using human-labeled annotation, we further investigate our CTID model. Figure 3.8 depicts the result of our additional analysis. Here, we compare the result of the CTID model with ground truth labeling. Using ground truth labeling the overall 8% performance is decreased.

Figure 3.9, depicts the result of the proposed model with the baseline model. Here, we compare the performance using ground-truth labeling. The overall performance of the baseline model also decreased. Here, we also find that the CTID model outperforms compared to the baseline model.

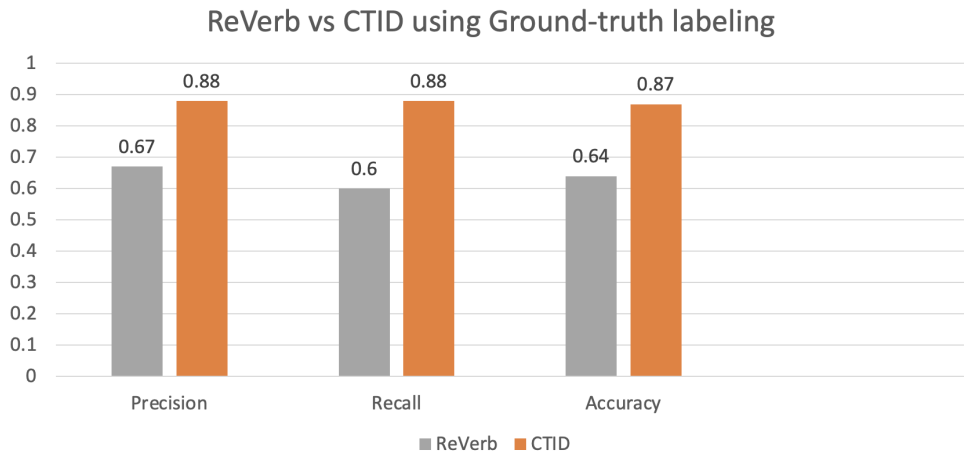

Figure 3.9: Experimental Result. Comparing the performance with the baseline model using ground truth labeling

3.7 Discussion

3.7.1 Additional Parameters

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.

3.7.2 Vocabulary Limitations

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.

3.8 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 study, 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.

4 Discussion

In this chapter, we discuss the assumptions of our research, the limitations of the CTID model with further analysis using ground truth labeling, and the main criteria for identifying competent triples.

4.1 Research Assumptions

4.1.1 Knowledge Extraction

In Open World Assumption (OWA), all entities and relations are not defined in the previous and there is no fixed format of information as most of the knowledge is in natural text format. Knowledge extraction for knowledge graphs under OWA also uses this assumption. In this research, our main focus is to identify knowledge from the natural text under OWA. As the OpenIE tool can extract triple-format data from text, we tried to utilize this data for knowledge graphs.

4.1.2 Truthfulness of Extracted Information

The main focus of our research is to retrieve competent triples from natural text data. Here, we utilize the OpenIE tool to extract triple from text automatically without any human supervision. The extracted triples by the OpenIE are not accurate in context. In this research, we use text data for creating our dataset and, here, we assume that all texts are true. As OpenIE takes natural text as input, it also does not provide any truthfulness about the extracted triples. Therefore, in the research work, the fact shown in any input sentence as well as in the extracted triples are assumed true. The context-based truthfulness of the extracted triples is out of the scope of this research. For example, "Barack Obama; is the president of; the USA" is a triple which is not currently accurate in context, but the format of the triple for being "Competent" is accurate. Here, we just ignore the context of the extracted triples. Our main focus is to retrieve "Competent" triples that are syntactically and semantically suitable for KGs.

4.1.3 Criteria of the Knowledge

Knowledge extraction for the KGs is the main focus of this research. Here, the extracted knowledge should have some criteria. As we utilize a QA dataset for creating our main data for the evaluation, we set some criteria for a triple to be a "Competent" one based on the available information in the QA dataset. The main criteria of the extracted knowledge are given below:

- The answer entity should be present in the triple. In the WebQuestionsSP dataset, the answer entity is given. While annotating the extracted triples, we utilize this answer entity to check whether it is present in the triple or not. However, as the triple's context is out of scope in this research, for ground truth labeling we did not strictly follow this criterion. For example, the question is "Who is the president of the USA?", the given answer entity is "Joe Biden", and the extracted triple is "Barack Obama; is the president of; the USA". Here, the answer entity is not present in the extracted triple but the structure of this triple for being a candidate is accurate. Therefore, this triple is labeled as "Competent".
- The triples should be semantically related to the corresponding question. For example, if the question is "What kind of money to take to the Bahamas?", the

answer entity is “Bahamian Dollar”, and the extracted triples are “The Bahamas; has; its own currency called the Bahamian dollar”, “All others; will dispense; Bahamian Dollar.”. Here, both triples contain the given answer entity but only the first one is semantically related to the given question. Therefore, the first triple is considered as “Competent” and the second triple is considered as “Incompetent” triple.

4.2 Limitations

4.2.1 Annotation Algorithm

For data annotation, we proposed an algorithm. After doing ground-truth labeling, we find some limitation in this algorithm which is discussed in section 3.6.

4.2.2 Semantic Similarity Problem in labeling

For ground truth labeling, we set two main criteria. Based on these criteria, human annotation is done. One of the criteria is to check the semantic similarity of extracted triple with the reference text. For this criterion, there is some space to think for humans which is not identical for all.

4.2.3 Necessity of Reference Text

We need reference text to identify competent triples from the triple set in our current settings. As we utilize a QA dataset, the questions are used as the reference text. Without reference text, the CTID model does not work.

4.2.4 Algorithm Selection

In this research, our main focus is to retrieve “Competent” triples from texts. The experiments conducted in our proposed CTID model are based on the classification model. Therefore, both the “Competent” and “Incompetent” classes are equally important for the model prediction, which affects our main focus. Using a model

which works only from positive examples may give better results rather than the classification model.

5 Conclusion

In this thesis, our main objective is to find competent triples from natural text data by utilizing openIE tool under open-world assumption. These triples can contribute to adding new information to the existing knowledge graphs. For this purpose, we proposed CTID model which includes two types of feature sets. One is syntax- and the other is semantic-based features. We investigate and evaluate our proposed features from different perspectives. Our model can contribute to finding competent triples from natural text which is ensure the quality of the extracted triples for the knowledge graph.

5.1 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 adding new information to the existing KGs. In this thesis, we propose the Competent Triple Identification (CTID) model for KGs. We also propose two types of features, 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 features 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.

5.2 Main Contribution

In the CTID model, 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. This automated annotation process is just a heuristic algorithm. This algorithm utilizes the existing answer entity from the QA dataset. By this heuristic approach, we can get the annotation for labeling data. But once we get the CTID model using this, we could apply our model without the dataset. 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.

5.3 Future Work

In the future, we will try to add these triples to complete existing KGs. Using these competent triples, we also can build openKG which is also a future plan from this research.

Bibliography

[1] World Wide Web Consortium et al. Rdf 1.1 concepts and abstract syntax, 2014. URL https://www.w3.org/TR/rdf11-concepts/. [online; accessed 4-May-2023].

[2] Wikipedia. Barack obama — wikipedia, the free encyclopedia, 2023. URL https://en.wikipedia.org/w/index.php?title=Barack_Obama&oldid=773703612. [online; accessed 4-May-2023].

[3] DBpedia. About: Barack obama, 2023. URL https://dbpedia.org/page/Barack_Obama. [online; accessed 4-May-2023].

[4] Xin Rong. word2vec parameter learning explained. In *Computing Research Repository (CoRR) abs/1411.2738*, 2014.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.

[6] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge Graph Embedding by Translating on Hyperplanes. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 1112–1119, 2014.

[7] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, page 2071–2080, 2016.

[8] 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*, pages 1–6, 2015.

[9] M Banko, MJ Cafarella, S Soderland, M Broadhead, and O Etzioni. Open

Information Extraction from the Web. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, page 2670–2676, 2007.

[10] Luciano Del Corro and Rainer Gemulla. Clausie: Clause-based Open Information Extraction. In *Proceedings of the 22nd international conference on World Wide Web*, pages 355–366, 2013.

[11] Kiril Gashteovski, Rainer Gemulla, and Luciano del Corro. Minie: Minimizing Facts in Open Information Extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. ACL, 2017.

[12] Lisa Ehrlinger and Wolfram Wöß. Towards a definition of knowledge graphs. In *Proceedings of the annual of SEMANTiCS conference (Posters and Demos)*, 2016.

[13] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. In *Proceedings of the IEEE, 104(1):11–33*, Jan 2016.

[14] Tim Berners-Lee. Linked data-design issues (2006), 2011. URL https://www.w3.org/DesignIssues/LinkedData.html. [online; accessed 4-May-2023].

[15] Douglas B. Lenat. Cyc: A large-scale investment in knowledge infrastructure. 38(11):33–38, November 1995, ISSN 0001-0782, URL https://dl.acm.org/doi/10.1145/219717.219745.

[16] Wikipedia. Cyc — wikipedia, the free encyclopedia, 2017. URL https://en.wikipedia.org/w/index.php?title=Cyc&oldid=773238151. [online; accessed 4-May-2023].

[17] George Miller. Wordnet: a lexical database for english. 38(11):39–41, November 1995.

[18] Juan Lloréns Jorge Morato, Miguel Angel Marzal and José Moreiro. Wordnet applications. In *Proceedings of Global WordNet Conference*, pages 20–23, 2004.

[19] Olivier Bodenreider. The unified medical language system (umls): integrating biomedical terminology. 32:D267–D270, 2004.

[20] Douglas Lenat. Understanding computers: Artificial intelligence. 1986.

[21] Fernando González-Ladrón-de-Guevara Enrique Estellés-Arolas. Towards an integrated crowdsourcing definition. *Journal of Information Science*, 38(2):189–200, 2012.

[22] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management*

of Data (SIGMOD), pages 1247–1250, 2008.

[23] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

[24] Thomas Pellissier Tanon, Denny Vrandečić, Sebastian Schaffert, Thomas Steiner, and Lydia Pintscher. From freebase to wikidata: The great migration. In *Proceedings of the 25th International Conference on World Wide Web (WWW)*, pages 1419–1428, 2016.

[25] Georgi Kobilarov-Jens Lehmann-Richard Cyganiak Sören Auer, Christian Bizer and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *Proceedings of the 6th International the Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference*, pages 722–735, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 3-540-76297-3, 978-3-540-76297-3.

[26] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195, 2015.

[27] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. Dbpedia-a crystallization point for the web of data. *Journal of Web Semantics*, 7(3):154–165, 2009.

[28] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 194:28–61, 2013. Artificial Intelligence, Wikipedia and Semi-Structured Resources.

[29] GeoNames. Geonames, 2023. URL http://www.geonames.org/. [online; accessed 4-May-2023].

[30] Ian Niles and Adam Pease. Towards a standard upper ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems*, page 2–9. ACM, 2001.

[31] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Jr. Estevam R. Hruschka, and Tom M. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, page 1306–1313. AAAI Press, 2010.

[32] Sebastian Padô Isabelle Augenstein and Sebastian Rudolph. Lodifier: Generating

linked data from unstructured text. In *Proceedings of the 9th International Conference on the Semantic Web: Research and Applications*, pages 210–224. Springer-Verlag, 2012.

[33] Hans Kamp and Uwe Reyle. *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. International series of monographs on physics. Springer Dordrecht, 1993.

[34] James R. Curran, Stephen Clark, and Johan Bos. Linguistically motivated large-scale nlp with c&c and boxer. page 33–36. Association for Computational Linguistics, 2007.

[35] Princeton University. Wordnet 3.0 in rdf. URL https://semanticweb.cs.vu.nl/lod/wn30/. [online; accessed 4-May-2023].

[36] Francesco Corcoglioniti, Marco Rospocher, Roldano Cattoni, Bernardo Magnini, and Luciano Serafini. *The KnowledgeStore: A Storage Framework for Interlinking Unstructured and Structured Knowledge*, pages 686–721. 2018.

[37] Vincent Kríž, Barbora Hladka, Martin Nečaský, and Tomáš Knap. Data extraction using nlp techniques and its transformation to linked data. pages 113–124, 2014.

[38] Martin Nečaský, Tomáš Knap, Jakub Klímek, Irena Holubová, and Barbora Hladka. Linked open data for legislative domain - ontology and experimental data. pages 172–183, 2013.

[39] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. page 601–610. Association for Computing Machinery, 2014.

[40] Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James Curran. Evaluating entity linking with wikipedia. *Artificial Intelligence*, 194:130–150, 2013.

[41] Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. Distant supervision for relation extraction without labeled data. pages 1003–1011, 2009.

[42] Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, page 2670–2676, 2007.

[43] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. pages 1535–1545, 2011.

[44] Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni.

Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534. Association for Computational Linguistics, 2012.

[45] Peter Exner and Pierre Nugues. Entity extraction: From unstructured text to dbpedia rdf triples. 2012.

[46] L. Galarraga, C. Teflioudi, K. Hose, and F. M. Suchanek. Exploring and evaluating attributes, values, and structures for entity alignment. In *Proceedings of the Very Large Data Bases*, page 707–730, 2015.

[47] Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. Fast rule mining in ontological knowledge bases with amie+. *The VLDB Journal*, 2015.

[48] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of NIPS*, pages 2787–2795, 2013.

[49] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of AAAI*, volume 14, pages 1112–1119, 2014.

[50] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of AAAI*, volume 15, pages 2181–2187, 2015.

[51] Han Xiao, Minlie Huang, and Xuan Zhu. Transg : A generative model for knowledge graph embedding. 2016.

[52] Alfred Horn. On sentences which are true of direct unions of algebras. *The Journal of Symbolic Logic*, 16(01):14–21, 1951.

[53] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of Advances in neural information processing systems*, pages 3111–3119, 2013.

[54] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In *In Workshop on Challenges in Representation Learning, ICML*, 2013.

[55] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In

Computing Research Repository (CoRR) abs/1412.3555, 2014.

[56] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. volume 9, pages 1735–1780, 1997.

[57] Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. volume 12, pages 2451–2471, 2000.

[58] Zhao Yan, Duyu Tang, Nan Duan, Shujie Liu, Wendi Wang, Daxin Jiang, Ming Zhou, and Zhoujun Li. Assertion-based qa with question-aware open information extraction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[59] Tushar Khot, Ashish Sabharwal, and Peter Clark. Answering complex questions using open information extraction. pages 311–316, 2017.

[60] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. Paraphrase-driven learning for open question answering. 2013.

[61] Amina Kadry and Laura Dietz. Open relation extraction for support passage retrieval: Merit and open issues. 2017.

[62] Alexander Löser, Sebastian Arnold, and Tillmann Fiehn. The goolap fact retrieval framework. *Lecture Notes in Business Information Processing*, 2011.

[63] Yuan Fang, Kingsley Kuan, Jie Lin, Cheston Tan, and Vijay Chandrasekhar. Object detection meets knowledge graphs. 2017.

[64] Gabor Angeli, Melvin Premkumar, and Christopher Manning. Leveraging linguistic structure for open domain information extraction. 2015.

[65] Niranjan Balasubramanian, Stephen Soderland, Mausam Mausam, and Oren Etzioni. Generating coherent event schemas at scale. 2013.

[66] Marco Ponza, Luciano Del Corro, and Gerhard Weikum. Facts that matter. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018.

[67] Xueling Lin, Haoyang Li, Hao Xin, Zijian Li, and Lei Chen. Kbpearl: a knowledge base population system supported by joint entity and relation linking. *Proceedings of the VLDB Endowment*, 2020.

[68] Travis Wolfe, Mark Dredze, and Benjamin Durme. Pocket knowledge base population. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 305–310, 2017.

[69] Federico Nanni, Jingyi Zhang, Ferdinand Betz, and Kiril Gashteovski. Eal: A toolkit and dataset for entity-aspect linking. In *Proceedings of the Joint Conference*

on Digital Libraries (JCDL), pages 430–431, 2019.

[70] Swapnil Gupta, Sreyash Kenkre, and Partha Talukdar. CaRe: Open knowledge graph embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 378–388. Association for Computational Linguistics, 2019.

[71] Samuel Broscheit, Kiril Gashteovski, Yanjie Wang, and Rainer Gemulla. Can we predict new facts with open knowledge graph embeddings? a benchmark for open link prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2296–2308, 2020.

[72] Denny Vrandečić and Markus Krötzsch. Wikidata: A Free Collaborative Knowledge Base. *Communications of the ACM*, pages 78–85, 2014.

[73] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia. *Artificial Intelligence*, 194:28–61, 2013.

[74] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: A Collaboratively Created Graph Database For Structuring Human Knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 1247–1250, 2008.

[75] David Liben-Nowell and Jon Kleinberg. The Link-prediction Problem for Social Networks. *Journal of the American Society for Information Science and Technology*, pages 1019–1031, 2007.

[76] Yankai Lin, Zhiyuan Liu, Maosong 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*, page 2181–2187, 2015.

[77] Takuma Ebisu and Ryutaro Ichise. Toruse: Knowledge graph embedding on a lie group. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[78] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. AMIE: Association Rule Mining under Incomplete Evidence in Ontological Knowledge Bases. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 413–422, 2013.

[79] Takuma Ebisu and Ryutaro 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*, pages 988–997, 2019.

[80] Heiko Paulheim and Christian Bizer. Improving the Quality of Linked Data using Statistical Distributions. *International Journal on Semantic Web and Information Systems*, 10:63–86, 2014.

[81] Lihua Zhao, Rumana Ferdous Munne, Natthawut Kertkeidkachorn, and Ryutaro Ichise. Missing RDF Triples Detection and Correction in Knowledge Graphs. In *Proceedings of the 7th Joint International Semantic Technology Conference*, page 164–180. Springer, 2017.

[82] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying Relations for Open Information Extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. ACL, 2011.

[83] Michael Schmitz, Robert Bart, Stephen Soderland, Oren Etzioni, et al. Open Language Learning for Information Extraction. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534. ACL, 2012.

[84] Natthawut Kertkeidkachorn and Ryutaro Ichise. T2KG: An End-to-End System for Creating Knowledge Graph from Unstructured Text. In *Proceedings of AAAI Workshop on Knowledge-based Techniques for Problem Solving and Reasoning*, page 743–749, 2017.

[85] Isabelle Augenstein, Sebastian Padó, and Sebastian Rudolph. LODifier: Generating Linked Data from Unstructured Text. In *Proceedings of the 9th International Conference on Semantic Web: Research and Applications*, pages 210–224. Springer, 2012.

[86] Vincent Kríž, Barbora Hladká, Martin Nečaskỳ, and Tomáš Knap. Data Extraction Using NLP Techniques and Its Transformation to Linked Data. In *Proceedings of the Mexican International Conference on Artificial Intelligence*, pages 113–124. Springer, 2014.

[87] Harith Alani, Sanghee Kim, David E Millard, Mark J Weal, Wendy Hall, Paul H Lewis, and Nigel R Shadbolt. Automatic Ontology-based Knowledge Extraction from Web Documents. *IEEE Intelligent Systems*, pages 14–21, 2003.

[88] Alexander Yates, Michele Banko, Matthew Broadhead, Michael Cafarella, Oren

Etzioni, and Stephen 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*, pages 25–26. ACL, 2007.

[89] Fei Wu and Daniel S. Weld. Open Information Extraction Using Wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 118–127. ACL, 2010.

[90] Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe: Global Vectors for Word Representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, 2014.

[91] Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina 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*, pages 201–206. ACL, 2016.

[92] Janara Christensen, Stephen Soderland, and Oren Etzioni. An Analysis of Open Information Extraction based on Semantic Role Labeling. In *Proceedings of the 6th International Conference on Knowledge Capture*, pages 113–120, 2011.

[93] Harinder Pal and Mausam. Demonyms and Compound Relational Nouns in Nominal Open IE. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, pages 35–39, 2016.