Automated Graph Generation at Sentence Level for Reading Comprehension Based on Conceptual Graphs

Wan-Hsuan Lin **Chun-Shien Lu**
Research Center for Information Technology Innovation
Institute of Information Science
Academia Sinica
Taipei, Taiwan, ROC
{rako1984@citi, lcs@iis}.sinica.edu.tw

Abstract

This paper proposes a novel miscellaneous-context-based method to convert a sentence into a knowledge embedding in the form of a directed graph. We adopt the idea of *conceptual graphs* to frame for the miscellaneous textual information into conceptual compactness. We first empirically observe that this graph representation method can (1) accommodate the slot-filling challenges in typical question answering and (2) access to the sentence-level graph structure in order to explicitly capture the neighbouring connections of reference concept nodes. Secondly, we propose a task-agnostic semantics-measured module, which cooperates with the graph representation method, in order to (3) project an edge of a sentence-level graph to the space of semantic relevance with respect to the corresponding concept nodes. As a result of experiments on the QA-type relation extraction, the combination of the graph representation and the semantics-measured module achieves the high accuracy of answer prediction and offers human-comprehensible graphical interpretation for every well-formed sample. To our knowledge, our approach is the first towards the interpretable process of learning vocabulary representations with the experimental evidence.

1 Introduction

In the ideal world, a knowledge representation technique, which solves the gamut of reading comprehension tasks from question answering to university entrance examinations, shows the ubiquity and understanding in all the corpora. In fact, for machine reading comprehension, developing a ubiquitous and well-defined knowledge representation which can be applied to open domain is still thorny in practice.

From the aspects of linguistics and neurophysiology, the study of syntax-semantics interface (Hackl, 2013) explains the effect of syntactic properties on semantic interpretation. That is why the syntax-semantics interface is of paramount importance for reading comprehension. In particular, the extraction of syntactic relations between the verb and its arguments plays a crucial role in the comprehension of language (Friederici and Weissenborn, 2007). Similarly, the abstract hierarchical syntactic structure of language is considered imperative for on-line comprehension even in a task-free environment (Brennan et al., 2016).

As one of methods which integrate the syntactic-semantic information, our graph generation model gains a better insight into the relationship between language processing and graph theory. In general, graph representation approaches regarding the syntax-semantics interface, are more suitable for the analyses of the relations between open-domain reading comprehension and hierarchical structure of syntax. In our case, we have the advantage in processing textual data effectively due to the combination of the theories of syntax-semantics interface and the practice of graph generation. By making use of hierarchical syntactic structure, our context-based graph model is beneficial to find the interactions and meanings between words; particularly, it is the fact that graphs capture interactions more efficiently between individual units of contextual data (Hamilton et al., 2017).

Proceedings of the 28th International Conference on Computational Linguistics, pages 2665–2675
Barcelona, Spain (Online), December 8-13, 2020

Drawing our inspiration from *conceptual graphs* (Sowa, 2000), we propose a miscellaneous-context-based method which allows a single sentence to be converted into a directed graph based on them, due to these logic-designed knowledge representations appropriate for semantic interpretation, higher-order expressions and metalevel operations of natural languages. Improving the reading comprehensibility, our mechanism generates a semantic network comprising the concept entities as graphical units for each sentence, and thence a piece of integrated information about each extracted concept entity is independently accessible to computer systems and comprehensible to human beings. Utilising the hypernym hierarchies provided by *WordNet* (Miller, 1998), we therefore project a sentence-level network to a large-scale lexical ontology for more explicit semantic positioning of concept entities, due to the fact that *WordNet* is designed by psycholinguistic features as well.

To our knowledge, our approach is the first towards the interpretable process of learning vocabulary representations with the experimental evidence. Here comes the emphases upon our contributions:

- Inspired by the design of conceptual graphs, we propose a novel miscellaneous-context-based method to address the open-domain reading comprehension by graph representations.

- Motivated by the syntax-semantics interface, we propose an algorithm to interpret the process of dismantling the sentence-to-graph conversion step by step.

- Evaluated on Wikipedia-collected datasets in terms of question answering, we demonstrate our graph representation approach and the semantics-measured module, which are free from training, achieve higher accuracy of answer prediction about relation extraction than the other models based on word embeddings.

2 Related Work

In the early stage of machine reading comprehension, a system (Hirschman et al., 1999) was first developed in usage of pattern matching techniques augmented with linguistic processing, for question-answering tasks. Following the trend of machine learning, a large-scale training dataset and a deep neural network cooperate on confronting the challenges of reading documents with less prior knowledge of language structure (Hermann et al., 2015). Following in the context of flourishing machine learning techniques, the DrQA model (Chen et al., 2017) and the Reinforced Ranker-Reader system (Wang et al., 2018) demonstrate the machine reading at scale through multitask learning on a variety of question-answering tasks. MQAN (McCann et al., 2018) without relying on any task-specific modules or parameters, demonstrates the improvement in transfer learning for a wide diversity of NLP tasks, such as machine translation, natural language inference and semantic parsing, etc.

Graphs are beneficial for the access to the structure of human knowledge. A graph database, known as Freebase (Bollacker et al., 2008), has relatively more definite data of general knowledge about types and properties, despite the limited capacity for knowledge expansion. As for question-answering tasks, the GTCR (Sun et al., 2018) and KG-MRC systems (Das et al., 2019) both predict the answers, involving temporal and causal relations, through the generation of specific knowledge graphs. For multi-hop reading comprehension, which requires multiple pieces of evidence to answer a question correctly, MHQA (Song et al., 2018) introduces the graph convolutional network (GCN) and graph recurrent network (GRN) to integrate and connect global evidence better.

Recently, graph representations gradually appear to address a wide range of reading comprehension challenges. For example, GTCR (Sun et al., 2018) generates event graphs for answering the temporal and causal questions, KG-MRC (Das et al., 2019) builds dynamic knowledge graphs for tracking the evolving states of participant entities, and the other variations of knowledge graph embeddings (Dettmers et al., 2018; Xiong et al., 2017; Cai and Wang, 2018; Xu et al., 2020; Vashishth et al., 2020; Sun et al., 2019; Zhang et al., 2019) have been presented.

3 Proposed Methods

In view of applications to the open domain of corpora, we bind a knowledge representation to generalisation of arbitrary well-formed text (*i.e.,* the text conforms the *grammatical well-formedness* developed

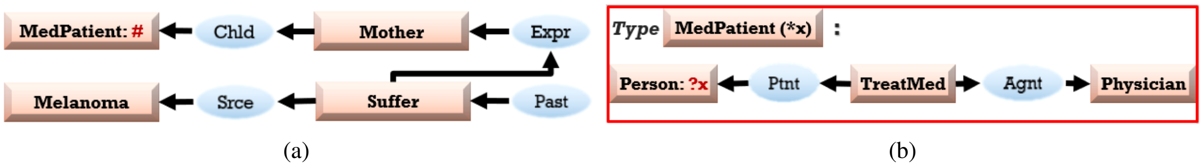

Figure 1: (a) A conceptual graph of the sentence *"The patient's mother suffered from melanoma"*. (b) The definition of the type *MedPatient*.

in generative linguistics.), for the purpose of dealing with reading comprehension issues. To this end, we propose a new method to automatically convert a sentence into a directed graph based on conceptual graphs, which provides the comprehensible format for not only computer systems but also human beings. Unlike most methods in exploiting machine learning techniques for feature extraction, this graphical knowledge embedding model consists of feature capture, synthesis, inference, tree-based structure forming, and semantic role labelling.

3.1 Preliminary: Conceptual Graphs

This paper focuses on a graph-based knowledge representation through sentence-level language processing. In view of language understanding, conceptual graphs are the best options of knowledge representations to meet our requirements, because their strict generative rules, compact top-level ontology and explicit notations are based on theoretical and computational linguistics. Next, we begin with a brief introduction of conceptual graphs:

- A conceptual graph consists of boxes and ovals, called *concepts* and *conceptual relations*, respectively. (See an example in Figure 1a)

- A concept node consists of a type label and an extension field, called a *concept type* and a *referent*, respectively. A referent can include a *quantifier* or an *indexical* symbol, which refer to the quantity and the current instance of a concept, respectively. If no other quantifier is specified inside a referent field, the default quantifier is the *existential* $\exists$ (Sowa, 2000). For example, in Figure 1a, [MedPatient: #], [Mother] and [Melanoma] have the *existential* quantifiers. If there is an indexical symbol # inside a referent field, it could be resolved to a specification of some instance with respect to its type label (Sowa, 2000). For example, [MedPatient: #] of Figure 1a refers to the type of person who receives the medical treatment by a physician, as shown in Figure 1b.

- A conceptual relation node represents the *case relation* (or thematic role) through linking the concept of a verb to one of a participant. For example, Figure 1a shows the concept [Suffer] has the experiencer (Expr) as [Mother] and the source (Srce) as [Melanoma].

3.2 Automated Graph Generation Based on Conceptual Graphs (AGG-CG)

Our model AGG-CG, called Automated Graph Generation based on Conceptual Graphs, aims for the analysis and generalisation of the sentences which satisfy the *grammatical well-formedness*, excluding the ill-formed information in an inappropriate context. It consists of two components, as illustrated in Figure 2: the node update process that classifies tokens in linear order with miscellaneous context-based features, as well as the edge connection part which determines the edges of a graph by the sentence's syntactic structure and captures case relations between nodes.

3.2.1 AGG-CG Part 1: Node Update

Given a well-formed input sentence $S = (s_1, s_2, \cdots, s_n)$ composed of n tokens s_i, we construct two sequences of nodes based on the definition of conceptual graphs (refer to the section 3.1). Two sequences of nodes are determined by the token categorisation of two word types: *content words* and *function words*. We only consider the type of content words, and assign nouns/verbs to a *actuality* node sequence $\mathbf{N_{actu}} = (N_1^{actu}, N_2^{actu}, \cdots, N_n^{actu})$ and adjectives/adverbs to a *characteristic* node sequence $\mathbf{N_{char}} =$

$(N_1^{char}, N_2^{char}, \cdots, N_m^{char})$. Limited to the tool usage of Stanford CoreNLP Toolkit (Manning et al., 2014) and *WordNet* (Miller, 1998), we implement the node update process supported by NLTK (Loper and Bird, 2004). If an input sentence has the potential compound verbs but does not be well-detected by these mentioned tools, we perform an extra analysis of compound verbs on a sentence before the token categorisation of word types. After the word type classification, tokens are embedded into concept node structure with the other information. The two-type concept node embeddings N_i^{actu} and N_i^{char} are in quintuple form $(\mathbf{P}, \mathbf{I}, \mathbf{W}, \mathbf{CT}, \mathbf{RF})$, where $\mathbf{P}$: part of speech tag, $\mathbf{I}$: position in the sentence, $\mathbf{W}$: the word token, $\mathbf{CT}$: concept type, and $\mathbf{RF}$: referent field. We process each node in linear order (*i.e.*, following the order of a sentence sequence) to update its concept type and referent field; moreover, reducing the redundant nodes is also imperative in this section.

The main objective of node update process is to extract and deduce the concept type and the referent for each content-word-embedded nodes. Considering the miscellaneous features with their contextual characteristics, we update all nodes firstly on their referent fields and secondly on their concept types. First of all, we process the contents of referent fields for all nodes $N_i^{actu} \in \mathbf{N_{actu}}$, which are taken into consideration according to (a) **subtypes or instances** and (b) **quantifiers** of possible concept type labels:

- **instances/subtypes:** Through NER parsing and POS tagging provided by Stanford CoreNLP Toolkit, there are some word tokens $\mathbf{W}$ associated with nodes N_i^{actu} (or N_i^{char}),which have their corresponding NER labels. We collect such words into a sequence by following their positions $\mathbf{I}$ relative to the context of a sentence and the next NER-labelled tokens, and then temporarily update the referent fields with either a sequence of NER-labelled words or an indication of empty. Next, we consider two different situations that a filled referent field will indicate either (i) an instance or a proper name of some concept entity or (ii) a subtype entity at lower level of a type hierarchy. Considering the context of various chunked NER-labelled word sequences with respect to the whole structure of a sentence, we determine a set of proper nouns or instances to fill the continuous referent fields by the segmentation of sequences separated into NER-correlated neighbouring nodes. Instead of complex proper nouns, the collection from the remaining NER-labelled words is appropriate for determining the single-word referents. On the other hand, a word token $\mathbf{W}$ is directly designated as a subtype entity in a referent field, if it is commonly acknowledged as a hyponym of the general type, such as colours and shapes. Regarding the processing of hyponyms/hypernyms, we usually analyse the subtrees of *WordNet* according to a word of a certain node.

- **quantifiers:** Through NER parsing, POS tagging, and government-dependant relations provided by Stanford CoreNLP Toolkit, we seek out a set of possible quantifiers and their corresponding objects (usually a word token $\mathbf{W}$ of a node). These possible quantifiers are generally presented as articles, determiners and numbers. By the definitions in *WordNet* and conceptual graphs, we assign Arabic numerals for articles/numbers, denote the *indexical* symbol for determiners, and remain the empty field for *existential quantifier*. This process not only allows us to observe all the quantifiers contextually but also provide information about the decision-making between subtypes and quantifiers.

Next, we determine the concept types which correlates with the previous analysis of referent fields. There are four cases of designating labels from different sources or eliminating concept nodes directly:

- **NER labels:** If a node's referent is an instance or a proper name, there will be a collection of NER labels corresponding to the word tokens in the referent field. (*i.e.*, the extraction of NER labels from a NER-labelled word sequence). We choose the label with the highest frequency of occurrences from the such collection, and determine it as the concept type.

- **hypernyms in *WordNet*:** If a node's referent is a subtype, we choose an appropriate hypernym as the concept type.

- **base form of word tokens:** If there is an arbitrary quantifier in the referent field, we acquire the base form of a certain node's word token and determine it as the concept type.

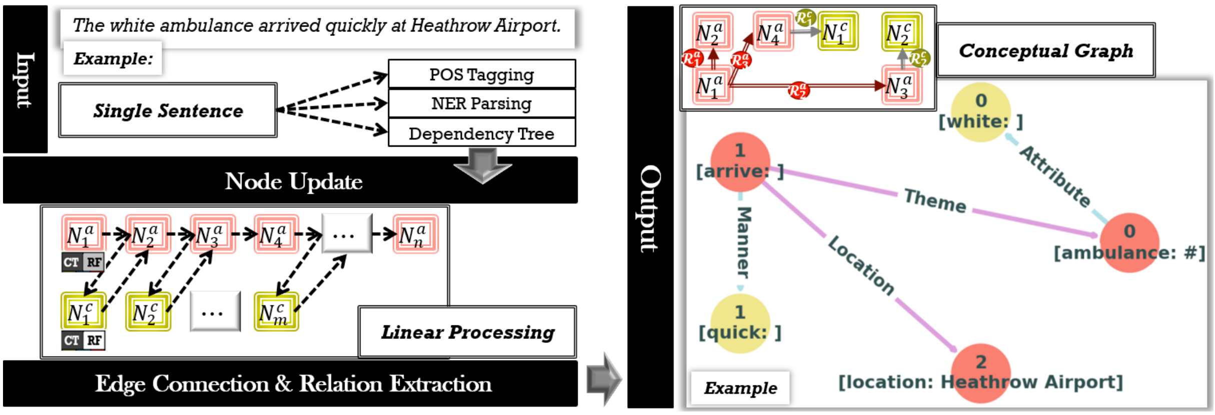

Figure 2: Overview of our graph representation model AGG-CG.

- **elimination:** If the certain node's referent field has no information about instances, subtypes and quantifiers, with the fact its word token $\mathbf{W}$ occurs as the part of the other node's referent, we eliminate this node from a set of vertices of a graph.

3.2.2 AGG-CG Part 2: Edge Connection & Relation Extraction

Conceptual relations represent the thematic roles or case relations between two concept nodes. There are two kinds of relations between concept entities: the first one links an act-associated concept (a verb-embedded N_i^{actu}) to a participant-associated concept (a noun-embedded N_j^{actu}), defined in the *KR* type *Participant*; and the second is usually revealed as a noun-to-noun, noun-to-adjective or verb-to-adverb dyadic relation of concept nodes, which is defined as the subtypes of the *KR* type *Actuality*. Please note that *Participant* (*i.e.*, the spatially distinguished parts of an *Occurrent*) and *Actuality* (*i.e.*, a physical entity with the independent existence), in addition to *Occurrent* (*i.e.*, an entity without a stable identity during any interval of time), are defined in the *KR Ontology* (Sowa, 2000). We denote a set of relations between two nodes in $\mathbf{N_{actu}}$ as $\mathbf{R_{actu}} = \{\mathcal{R}_1^{actu}, \mathcal{R}_2^{actu}, \cdots, \mathcal{R}_l^{actu}\}$, and a set containing the relations between nodes N_i^{actu} and N_j^{char} as $\mathbf{R_{char}} = \{\mathcal{R}_1^{char}, \mathcal{R}_2^{char}, \cdots, \mathcal{R}_k^{char}\}$. To explore what the structure of existing concept nodes forms, we utilise the dependency tree of a sentence to generalise the base of structure, and afterwards the linguistic patterns or rules are adopted to enhance the structural completeness. Through the complex analysis on connections between concept nodes, the more stable architecture of a graph is founded, usually leading to the less thorny determination of the relation types for each connection. In order to obtain more information about relations between concept nodes, we refer to lexical units corresponding to concept types of act-associated nodes, which are provided by *FrameNet* (Baker et al., 1998). When we find a proper frame for each act-associated node N_i^{actu}, we choose the labels of frame elements as *reference labels* which have close lexical similarity with concept types of the other participant-associated nodes N_j^{actu}, and take these reference labels into consideration to narrow down the list of relation types defined in type *Participant* of *KR Ontology*. It seems a way of building bridges between *FrameNet* and *KR Ontology*. At the final stage, we reduce the unused nodes and expand the isolated ones, to contextualise the more complete structure of a graph output.

3.3 The Semantics-measured Module (SMM)

Because our graph representation method lacks a metric to define each pair of concept entities in mathematical expression, we design a simple semantics-measured module (SMM) to map the extracted concept nodes to the *WordNet* hypernym hierarchy and compute the semantic similarities between two nodes. SMM attempts to achieve a proper projection across the interface between syntax and semantics.

A sample of question-answering tasks contains a *Question* and a set of *Statements*. The functions of our semantics-measured module, as shown in Figure 3, are to (I) extract the concept nodes at the fixed levels from a graph generated by *Statements* and (II) compute a semantic similarity of each pair of elements from a *candidate set* and a *reference set*, respectively. By definition, the *candidate set* is

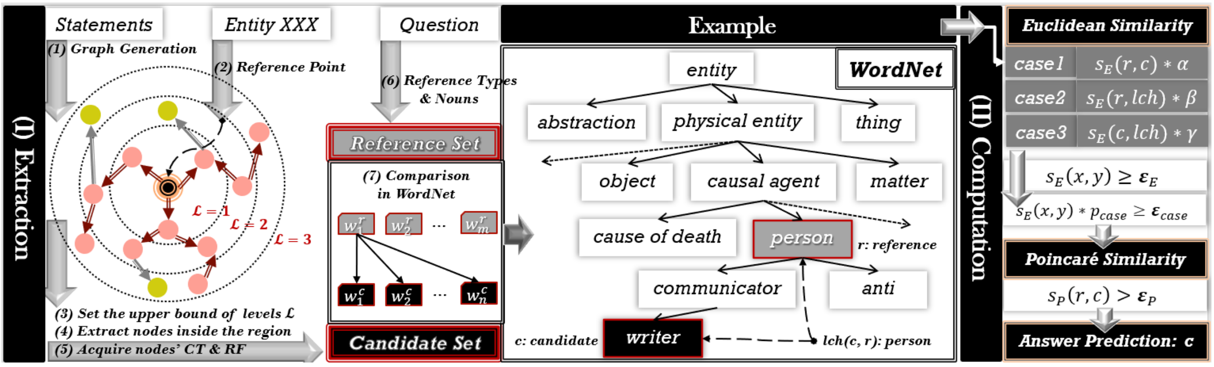

Figure 3: The structure of the semantics-measured module(SMM).

composed of concept types and referents of the concept nodes from *Statements*, and a *reference set* comprises the nouns and reference types associated with a *Question*. A sample of question-answering tasks contains a *Question* and a set of *Statements*.

We adopt the path distance similarity provided by NLTK project (Loper and Bird, 2004) in order that a semantic similarity between two nodes is intuitively understood by the shortest path connecting two projected concept entities in *WordNet*'s lexical hierarchy dependent on the is-a taxonomy. The definition of path distance similarity between two word senses $x, y \in X$ is:

$$s(x,y) = \frac{1}{1 + d(x,y)},$$

where $s : X \times X \to [0,1]$ is the score function with the distance $d : X \times X \to [0,\infty)$.

Inspired by the experimental facts of the Poincaré embeddings (Nickel and Kiela, 2017), which takes advantage of hyperbolic geometry for learning hierarchical representations, we also take the hyperbolic distance into consideration for semantic similarity computation. We adopt the two path distance similarities for prediction according to the highest score, where the similarity functions depend on the following two distance functions:

- **Euclidean:**

$$d_E(x,y) = \|x - y\|^2$$

- **Poincaré :**

$$d_P(x,y) = \operatorname{arcosh}(1 + 2\frac{\|x-y\|^2}{(1 - \|x\|^2)(1 - \|y\|^2)})$$

In the part of similarity computation, we firstly obtain the *Lowest Common Hypernym* (lch) of the certain pair of a *candidate* c and a *reference* r and secondly calculate the Euclidean path similarities $s_E(r,c)$, $s_E(r,lch)$, and $s_E(c,lch)$ among the three. According to the results of these computations, we go through the robin-round comparison between *candidate set* and *reference set*, and add similarity scores s_E to the collection $\mathcal{S}$ if they are verified through the following three cases in the current order:

case1: $s_E(r,c) \geq \max \mathcal{S}$, and lch is r or c.
case2: $s_E(r,lch) \geq \max \mathcal{S}$.
case3: $s_E(c,lch) \geq \max \mathcal{S}$.

Given the specific Euclidean threshold ε_E, case threshold ε_{case} and case parameter p_{case}, we assign the different $p_{case} = \alpha, \beta, \gamma$ to the $case = 1, 2, 3$ respectively, and manage to filter out more candidate entities with the constraints ε_E and ε_{case}. After each case determination of a tuple (r, c, lch), we verify whether the corresponding $s_E \in \mathcal{S}$ satisfies $s_E \geq \varepsilon_E$ and $s_E * p_{case} \geq \varepsilon_{case}$ or not, and eliminate the redundant ones from $\mathcal{S}$. To take the last step, we collect each pair (r,c) corresponding to a semantic similarity s_E in the current refined $\mathcal{S}$, and compute all the Poincaré path similarities of reference-candidate pairs $s_P(r,c)$ in this collection. Imposing the restriction $s_P(r,c) > \varepsilon_P$, we narrow down the list of candidate entities to a few prediction and choose the one with the highest Poincaré similarity score.

Models	Precision	Recall	F1
Random NE	9.25%	18.06%	12.23%
RNN Labeler	13.28%	5.69%	7.97%
End-to-End RE	100.00%	0.00%	0.00%
S2S (Multitask Training)	-	-	28.50%
MQAN (QA-first)	54.78%	28.67%	37.64%
MQAN (Sampling + CoVe)	56.80%	32.00%	40.94%
***Five variants** of the model (Levy et al., 2017)*			
KB Relation	19.32%	2.54%	4.32%
NL Relation	40.50%	28.56%	33.40%
Single Template	37.18%	31.24%	33.90%
Multiple Template	43.61%	36.45%	39.61%
Question Ensemble	45.85%	37.44%	41.11%
Our model			
AGG-CG+SMM	**52.78%**	**44.33%**	**48.19%**

Table 1: Results of relation extraction with unseen relations.

4 Experimental Results and Comparison

In this section, we evaluate the pipeline that attaches the graph representation AGG-CG to the module SMM on the question-answering-based task of relation extraction. According to the slot-filling data collected from Wikipedia, it allows us to demonstrate the ability of this graph representation method for narrowing down the candidates for the answer, through the access of graphs' hierarchical trees at a certain level.

4.1 Datasets

Experiments were conducted on the dataset in a zero-shot setting (abbreviated as ZRE here) for the task *Unseen Relations* (Levy et al., 2017). The ZRE dataset was collected from WikiReading (Hewlett et al., 2016), in which each sample consists of a relation label, a specific entity, an answer, a question and a set of statements. Following the same setting in (Levy et al., 2017), we only acquired the information of unannotated statements and questions as the materials for processing, in addition to specific entities as reference points for graphs.

4.2 Setup

In evaluation, we only adjusted the parameters in the module of semantic similarity computation. We set the upper bound of levels of candidate nodes $\mathcal{L} = 4$ by indicating a reference point as the root. The tolerance thresholds of Poincaré path similarity $\varepsilon_P \in \{\epsilon_o, \epsilon_s\}$ defined in *WordNet* and NLTK were set as $\epsilon_o = 0.63$ ordinarily, and as $\epsilon_s = 0.70$ in specific situation if a candidate is different from its corresponding node's concept type.

4.3 Comparison

According to Table 1, we compare our models with six other systems, in which one of the systems has five variants (Levy et al., 2017). Next, we go through and introduce the entries on Table 1 one by one.

As the baseline for relation extraction in (Levy et al., 2017), *Random NE* randomly chooses a name entity, which does not exist in a question, from a set of statements for each sample. *RNN Labeler* is a RNN model proposed in company with the WikiReading dataset (Hewlett et al., 2016), which operates on the sequence of words and estimates the coherence of the current word and the answer at each time step. *End-to-End RE* is referred to the end-to-end RNN-based model (Miwa and Bansal, 2016), which captures both word sequence and dependency tree substructure in order to extract entities and relations. *S2S* is a pointer-generator sequence-to-sequence model (See et al., 2017) for text summarisation, which

ϵ_o	ϵ_s	Precision	Recall	F1
0.53	*0.60*	50.38%	44.00%	46.98%
0.63	*0.70*	52.78%	44.33%	48.19%
0.73	*0.80*	52.31%	37.67%	43.80%
0.83	*0.90*	59.62%	31.00%	40.79%

(a)

$\mathcal{L}$	Precision	Recall	F1
2	53.55%	32.67%	40.58%
3	53.12%	39.67%	45.42%
4	52.78%	44.33%	48.19%
5	51.56%	44.00%	47.48%

(b)

Table 2: (a) The results of AGG-CG+SMM with different tolerance thresholds (ϵ_o, ϵ_s), given $\mathcal{L} = 4$. (b) The results of AGG-CG+SMM with different upper bounds $\mathcal{L}$, given $(\epsilon_o, \epsilon_s) = (0.63, 0.70)$.

is jointly trained on all tasks of the decaNLP challenge (McCann et al., 2018) by a round-robin batch-level sampling strategy. *MQAN* is the multitask question answering network (McCann et al., 2018) utilising the novel dual coattention and multi-pointer-generator decoder, which goes through a fully joint and round-robin training to multitask across the whole decaNLP and avoids the usage of task-specific modules or parameters. According to different training strategies, *MQAN (QA-first)* is directly trained on all the tasks of decaNLP by fully joint strategy, and *MQAN (Sampling+CoVe)* is pretrained on seven difficult tasks of decaNLP in the first phrase of anti-curriculum strategy. Moreover, *MQAN (Sampling+CoVe)* exhibits more impressive performance on decaNLP by making better use of context vectors (McCann et al., 2017).

Speaking of the model (Levy et al., 2017), it is modified from the Bi-Directional Attention Flow (BiDAF) network (Seo et al., 2017) which uses an attention mechanism to achieve a query-aware context representation. For experimental necessity, there are five variants of this relation-extraction model: (i) *KB Relation* only uses the statements and a relation indicator instead of a question as the reference materials. Apart from a set of statements, (ii) *NL Relation* merely uses the name of relation label per example instead of a question. (iii) *Single Template* indicates the usage of one single question template per relation label and (iv) *Multiple Template* adopts a variety of questions according to the same relation, during the training stage. For each sample in the dataset, (v) *Question Ensemble* queries a set of statements with three questions and allows the choice between three predictions, in the form of an ensemble.

Our model AGG-CG+SMM, combining the graph representation AGG-CG and the semantics-measured module SMM, does not need training.

4.4 Evaluation

The evaluations were directly performed on 10-fold test/dev set, where the numbers of positive and negative samples are approximately equal per fold. Adopting the same evaluation tool provided by zero-shot relation extraction experiments (Levy et al., 2017), we evaluated the model by the scores—Precision, Recall and F1—through comparing the tokens in the given answer with the prediction. By definition, precision is the number of true positives divided by the number of all non-null results retrieved by the system, recall is the number of true positive samples divided by the total number of positives, and F1 is the harmonic mean of precision and recall.

4.5 Results

Table 1 reveals the experimental results of all the systems' performance for relation extraction with unseen relation labels, where the empirical data of the other systems is directly cited from either (Levy et al., 2017) or (McCann et al., 2018). As the simply designed models, *Random NE*, *RNN Labeler*, *End-to-End RE* and *KB Relation* fail to detect the ground-truth answers due to the improper processing structure or the flawed representations of natural languages. As the comparative systems, both *Multiple Template* and *Question Ensemble* have access to more reference materials about multiple questions of datasets and therefore generate more exact predictions; similarly, *MQAN* has the more complete strategies to train on multiple natural language processing tasks. On the other hand, our system AGG-CG+SMM has achieved the highest accuracy in terms of *Recall* and *F1*, which is free from the reference materials of extra samples and multitask training on different datasets.

Models	Precision	Recall	F1
LOEE+SMM	22.09%	6.33%	9.84%
AGG-CG+SMM	52.78%	44.33%	48.19%

(a)

Models	EM	nF1
LOEE+ASEM	11.31%	47.11%
AGG-CG+ASEM	30.69%	62.93%

(b)

Table 3: (a) The impact of AGG-CG on the overall performance of relation extraction (ZRE). (b) The impact of AGG-CG on the overall performance of semantic role labelling (QA-SRL).

5 Analysis

In this section, we demonstrate how the parameter setup of SMM impacts the performance of the whole pipeline and how the conceptual graph representation acts as a critical role of different pipelines in natural language processing tasks.

5.1 Poincaré Tolerance Thresholds of SMM in Relation Extraction Task

Table 2a shows the performance of our pipeline through the adjustment of tolerance thresholds ϵ_o and ϵ_s of SMM, with the fixed upper bound of levels $\mathcal{L} = 4$. Apparently, recall scores are decreasing when the tolerance thresholds have a rise of amounts gradually. The case, which compares the *references* derived from *Question* with the *candidate set* derived from *Statement* within stricter constraints of their path similarities, will fail in solving positive examples more frequently. In contrast, the rates of precision increase in the same situation, due to more noticeable differentiation of negative examples among the total cases.

5.2 Upper Bound of Levels of SMM in Relation Extraction Task

Table 2b shows how the different upper bounds of levels $\mathcal{L}$ of SMM affect the accuracy of the model for fixed tolerance thresholds $\epsilon_o = 0.63$ and $\epsilon_s = 0.70$. The increasing value of $\mathcal{L}$ indicates the access to more concept nodes in chains extended from a reference point in a sentence-level graph. This means the upper bound $\mathcal{L}$ determines the cardinality of the *candidate set*. Naturally, from more candidates we can choose the correct answer for each positive example; however, the redundant candidate nodes are taken into consideration for processing negative examples.

5.3 Conceptual Graph Representation in Relation Extraction Task

For the relation extraction task, we empirically observed that conceptual graph representation has the effect on the performance of the whole pipeline. We design the baseline LOEE (Linear Ordering Entity Extraction), which sequentially converts entities of a sentence into nodes connected in linear ordering. Certainly, this baseline is suitable for the node processing and similarity computation.

Table 3a shows that the pipeline with AGG-CG's node representation tremendously outperforms the one with baseline LOEE, where this analysis has the same parameter setup of SMM as in Section 4.2. The result also claims that language processing in hierarchical structure is more beneficial than linearly processing sentences as sequences.

5.4 Conceptual Graph Representation in Semantic Role Labelling Task

In order to experimentally verify that AGG-CG is appropriate and effective to the other natural language processing tasks, we evaluated on the QA-SRL (He et al., 2015) in the domain of Wikipedia articles, where this task simply uses question-answer pairs to annotate thematic roles. The structure of QA-SRL dataset is very different from ZRE, particularly the forms of questions and ground-truth answers. In view of this fact, we demonstrate the flexibility and cooperativity of AGG-CG with the other node-processing modules. For QA-SRL, the Answer-Section Extraction Module (ASEM) of the pipelines is introduced to obtain an answer-section of a statement through node processing on representation. As a comparative baseline, we select the same LOEE as in Section 5.3.

Table 3b shows that conceptual graph representation is applicable to the other natural language processing tasks, and demonstrates the advantage of AGG-CG trough the performance. As a result of

QA-SRL evaluation, we know that LOEE+ASEM succeeded in meeting expectations, which assess the suitability of ASEM and eliminate the suspicion of the baseline's ineffectiveness on ZRE dataset.

6 Conclusion and Future Work

The main theme of this paper presents the graph method for knowledge representations to process textual data at sentence level, which is designed to tackle open-domain corpora but not serve as a task-oriented model for reading comprehension tasks. In future work, we will consider enhancing semantic role labelling between nodes and conceive an algorithm of graph composition between sentences at paragraph level, in order to combine the language with more complicated geometric analysis and graph theory. From the experimental aspect, we will demonstrate that our graph representation cooperating with a simple task-oriented module can be evaluated on more QA-type tasks, for presenting the ubiquity of AGG-CG representations on natural language understanding.

Acknowledgements

This work was supported by Ministry of Science and Technology, Taiwan, ROC, under Grants MOST 107-2221-E-001-015-MY2, MOST 109-2634-F-007-018, and MOST 109-2221-E-001–023.

References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet Project. In *Proc. of ACL 1998*, pages 86–90.

Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD Conference 2008)*, pages 1247–1250.

Jonathan R. Brennan, Edward P. Stabler, Sarah E. Van Wagenen, Wen-Ming Luh, and John T. Hale. 2016. Abstract linguistic structure correlates with temporal activity during naturalistic comprehension. *Brain and Language*, 157:81–94.

Liwei Cai and William Yang Wang. 2018. KBGAN: Adversarial learning for knowledge graph embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018)*, pages 1470–1480.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proc. of ACL 2017*, pages 1870–1879.

Rajarshi Das, Tsendsuren Munkhdalai, Xingdi Yuan, Adam Trischler, and Andrew McCallum. 2019. Building dynamic knowledge graphs from text using machine reading comprehension. In *7th International Conference on Learning Representations (ICLR 2019)*.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI 2018)*.

Angela D. Friederici and Jürgen Weissenborn. 2007. Mapping sentence form onto meaning: The syntax-semantic interface. *Brain Research*, 1146:50–58.

Martin Hackl. 2013. The syntax-semantics interface. *Lingua*, 130:66–87.

William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation learning on graphs: Methods and applications. *IEEE Data Engineering Bulletin*, 40(3):52–74.

Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proc. of Conf. on EMNLP 2015*, pages 643–653.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems (NIPS 2015)*, pages 1693–1701.

Daniel Hewlett, Alexandre Lacoste, Llion Jones, Illia Polosukhin, Andrew Fandrianto, Jay Han, Matthew Kelcey, and David Berthelot. 2016. Wikireading: A novel large-scale language understanding task over wikipedia. In *Proc. of ACL 2016*, pages 1535–1545.

Lynette Hirschman, Marc Light, Eric Breck, and John D. Burger. 1999. Deep read: A reading comprehension system. In *Proc. of ACL 1999*, pages 325–332.

Omer Levy, Min Joon Seo, Eunsol Choi, and Luke S. Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017).*

Edward Loper and Steven Bird. 2004. NLTK: The natural language toolkit. In *Proc. of ACL 2004*, pages 214–217.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proc. of ACL 2014*, pages 55–60.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems (NIPS 2017)*, pages 6294–6305.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730.*

George A Miller. 1998. *WordNet: An electronic lexical database*. MIT press.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proc. of ACL 2016*, pages 1105–1116.

Maximilian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems (NIPS 2017)*, pages 6338–6347.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proc. of ACL 2017*, pages 1073–1083.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *5th International Conference on Learning Representations (ICLR 2017).*

Linfeng Song, Zhiguo Wang, Mo Yu, Yue Zhang, Radu Florian, and Daniel Gildea. 2018. Exploring graph-structured passage representation for multi-hop reading comprehension with graph neural networks. *CoRR*, abs/1809.02040.

John F. Sowa. 2000. *Knowledge representation: logical, philosophical, and computational foundations*. P.W.S.

Yawei Sun, Gong Cheng, and Yuzhong Qu. 2018. Reading comprehension with graph-based temporal-casual reasoning. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*, pages 806–817.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations.*

Shikhar Vashishth, Soumya Sanyal, V. Nitin, N. Agrawal, and P. Talukdar. 2020. Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions. In *AAAI.*

Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerald Tesauro, Bowen Zhou, and Jing Jiang. 2018. R3: Reinforced ranker-reader for open-domain question answering. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI 2018).*

Wenhan Xiong, Thien Hoang, and William Yang Wang. 2017. DeepPath: A reinforcement learning method for knowledge graph reasoning. In *Proc. of Conf. on EMNLP 2017*, pages 564–573.

Chengjin Xu, M. Nayyeri, Y. Chen, and J. Lehmann. 2020. Knowledge graph embeddings in geometric algebras. *ArXiv*, abs/2010.00989.

Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019. Quaternion knowledge graph embedding. In *NIPS.*