applied sciences

MDPI

Article

GREG: A Global Level Relation Extraction with Knowledge Graph Embedding

Kuekyeng Kim, Yuna Hur, Gyeongmin Kim and Heuiseok Lim *

Department of Computer Science and Engineering, Korea University, Seoul 02841, Korea; overmind22@korea.ac.kr (K.K.); yj72722@korea.ac.kr (Y.H.); totoro4007@korea.ac.kr (G.K.)

* Correspondence: limhseok@korea.ac.kr; Tel.: +82-2-3290-2396

Received: 21 October 2019; Accepted: 30 January 2020; Published: 10 February 2020

Abstract: In an age overflowing with information, the task of converting unstructured data into structured data are a vital task of great need. Currently, most relation extraction modules are more focused on the extraction of local mention-level relations—usually from short volumes of text. However, in most cases, the most vital and important relations are those that are described in length and detail. In this research, we propose GREG: A Global level Relation Extractor model using knowledge graph embeddings for document-level inputs. The model uses vector representations of mention-level 'local' relation's to construct knowledge graphs that can represent the input document. The knowledge graph is then used to predict global level relations from documents or large bodies of text. The proposed model is largely divided into two modules which are synchronized during their training. Thus, each of the model's modules is designed to deal with local relations and global relations separately. This allows the model to avoid the problem of struggling against loss of information due to too much information crunched into smaller sized representations when attempting global level relation extraction. Through evaluation, we have shown that the proposed model yields high performances in both predicting global level relations and local level relations consistently.

Keywords: relation extraction; knowledge graph; meta learning; text summarization; natural language processing; machine learning

1. Introduction

Several machine learning applications, such as Natural Language Processing, are rapidly increasing its reliance on deep learning due to their high accuracy, performance, and adaptability [1]. Which resulted in the importance and volume of data to grow exponentially. However, most types of raw data remain in an unstructured format, making it difficult to interpret while displaying low coherency. Thus, converting unstructured information to structured information is a vital process within the field [2]. However, to do this process manually demands huge amounts of time and cost, and therefore the need to automate this process has arisen.

One of the most popular methods of converting unstructured data into a structured format is by relation extraction. To perform relation extraction, identification must be performed for each entity within the data. This is then followed by identifying whether each pair of entities are semantically related. Finally, through analysis, the identified semantic relations are classified accordingly. While several previous attempts performed toward smaller sets of text data have shown promising results, performances against larger sets of data tend to suffer. This shortcoming is especially easy to spot when having to identify global mention-level relations. Global mention-level relations are relations that stretch among long contexts, usually described through the course of many other intermediate sub-relations [3,4]. This kind of short-sightedness may also affect the accuracy of local relation extractions as well, since the model may misinterpret ambiguous relations or even omit them entirely if there are types of relations that only become coherent when viewed on a more global scale such as metaphors or sarcasm.

In this article, we propose a relation extraction method that focuses on extracting global relations by utilizing knowledge graph construction and processing. The model largely consists of two parts: a context-aware long short-term memory(LSTM) [5] based relation extraction module and a knowledge graph constructor module that produces knowledge graphs from the given document. By constructing a knowledge graph from the input document, the constructed knowledge graph is used to extract the input document's global relations through graph embedding. To do this, both modules go through a synchronization process. We achieved this by training the two modules in an alternating fashion. This allows the proposed model to extract and classify local relations while concerning its possible global relations. For this research, we have constructed a new set of data consisted of 10,470 lines of documented Korean text of history [6] for this task. The dataset was constructed purely through human annotations and contains the information of both local relations per each sentence and knowledge graphs that represents each document.

The following article will consist of the following: Section 2 to show previous or related methods as well as introduce what our improvements are. Section 3 will contain details of our proposed model's structure and process and how it was trained. In Section 4, we shall introduce our newly constructed dataset as well as comparisons from other models and analyze the results in detail, proving the viability of our proposed model. Finally, we shall summarize this article and speculate future works in Section 5.

2. Related Works

2.1. Relation Extraction

Relation Extraction can largely be divided into two types: global level relation extraction and mention-level relation extraction [7], while the former deals with 'global' level relations that span across multiple relations and large bodies of text, and mention-level 'local' relations are smaller scale relations expressed directly within short sentences.

The best example that shows the difference between the two can be shown by syllogism. Given two sentences "All cars have wheels.' and "A truck is a car", a local relation extraction will only be able to extract the information $(Car, possess, Wheel)$ and $(Truck, is, Car)$. Thus, it will never be able to extract and represent what the two sentences combined mean: "A truck has wheels". On the other hand, global relation extraction would consider these global level relations and would be able to extract the information $(Truck, Possess, Wheel)$.

Currently, most relation extraction modules that yield high performances usually only deal with 'local relations'. However, when representing a large amount of text such as a document, representations only through local relations are thoroughly limited. Considering the fact that the most valuable information within documents is those that have been described in detail and are spanned across the document, the importance of global level relation extraction is not something to be looked over so lightly.

However, the task itself is nowhere near simple. The main cause that prevents current state-of-the-art local relation extraction methods to properly perform global relation extraction is due to the bottleneck problem [8]. With so much information to process for a singular relation, the loss of information is inevitable—thus hindering the model's capability to properly judge the relation or capture it at all. On the other hand, if the model is focused only on extracting global relations, then, in an attempt to avoid the bottleneck problem [8], its performance against local relation extraction drops due to the sparsity of information [9]. Previous attempts on global relation extraction have tried several different methods to tackle this problem. For instance, Gerber et al. [10] attempted this by simply using co-references to gain access to other sentences without actually having to model them or Quirk et al. [11] who applied distant supervision to general global relation extraction as well.

One of the more recent approaches includes Peng et al. [3] attempting global relation extraction by continuously retaining the entity representations learned from the entire text as input, allowing the extracted information from the whole text to be easily extended to other sentences as well.

In this research, we propose a divide and conquer approach to the task. By using two separate modules to perform local relation extraction and global relation extraction separately, we entirely avoid the stated bottleneck problem [8] and the following dilemma. As global relations can largely be seen as an amalgamation of multiple local relations linked together, the proposed model aims to extract local relations through the first module and converts them into a knowledge graph. The constructed knowledge graph is then analyzed to extract global relations.

2.2. Knowledge Graph Embedding

Knowledge graphs, also known as ontologies, are a form of structured data represented through the entities and relations it is composed of [12]. As one of the most simple yet effective forms of representations, its compositions shares many traits with that of relation extractions, since they are both used to represent unstructured data in a structured form. However, a major difference between the two methods does exist: where the results from relation extractions are a one-shot instance that represents only a small portion of information, the result and representations from a knowledge graph constructor are expected to have consistency for every relation, context, and node it represents. This is because a knowledge graph represents a large portion of information in a structured fashion [13]. Thus, every representation from knowledge graphs must not be ambiguous regarding how they are presented, demanding consistency. This property is what allows knowledge graphs to be a reliable form of representation when representing large amounts of data—thus making them a viable option when trying to embed certain domains or documents in fields such as natural language processing [14], multimodal processing [15], etc. Inspired by this, the proposed model utilizes knowledge graph embeddings for the extraction of global relations from documents. The knowledge graph is constructed through local relations with its nodes as entities, links as relation classes, and each node's context as the vector representation of each entity and relation. Through the synchronization process, the resulted knowledge graph from the constructor is trained to retain their consistency in representations of both local and global relations while also containing accurate relation extraction information as well.

3. Model

The proposed model largely consists of two synchronized modules. While the first module is loosely based around a context-aware LSTM based relation extraction module that was proposed by Sorokin et al. [16], the second module is a knowledge graph constructor module with the given input as tuples of entities and their relations. The synchronization process is achieved by alternating the two modules during training.

The overall process of the module is as shown in Figure 1. The given text is first preprocessed to be divided into sentences which are then passed to a named entity recognition module [17,18] to identify the entities within the given text. After the identification, the relation extraction module is used to extract local relations from the identified entities within each sentence. Once all relations are extracted, they are used to create a knowledge graph of entities via knowledge graph constructor module. Through this constructor module, the loose relations of entities that are stretched within large bodies of texts are grouped, allowing for extracting global scale relations. To do this, the knowledge graph constructor is assigned to find entities that may act as possible hubs or bridges between other entities. In addition, by acknowledging the type of relations that are between the candidate entities as separate vectors, it will identify if the amalgamation of these vectors does indicate a relation. Once an amalgamation is required, the constructor classifies the global relation based on cosine vector similarity.

This indicates that our proposed global relation extraction method is highly dependent on the output from the local relation extraction. Thus, proper synchronization is essential since both modules must consider the other while training themselves. As seen in Figure 2, once the initial training of the local relation extraction module is finished, a set of documents from the training dataset is divided into sentences. These sentences are then given as input. This converts the set of documents composed of sentences into a set of documents composed of local relation triplets. Based on the given local relation triplets, the knowledge graph constructor module aims to construct a knowledge graph containing global and local relations accurately.

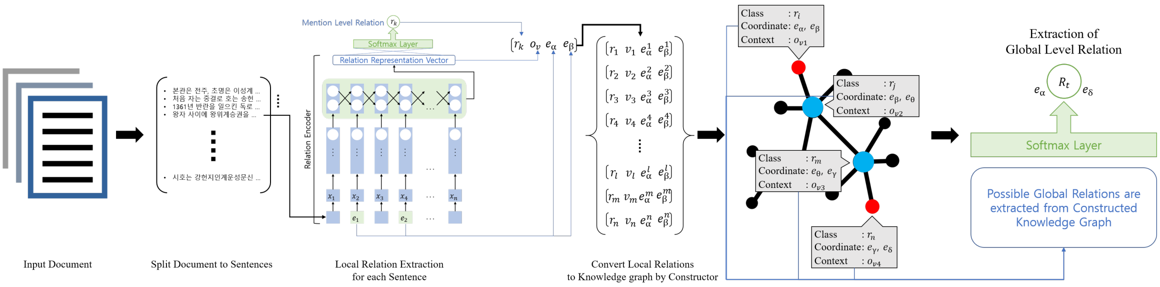
Mention Level Relation
r_k
Softmax Layer
Relation Representation Vector
$(r_k \ o_v \ e_\alpha \ e_\beta)$
본관은 전주, 초명은 이성계 ...
처음 자는 중결로 호는 송헌 ...
1361년 반란을 일으킨 독로 ...
왕자 사이에 왕위계승권을 ...
시호는 강헌지인계운성문신 ...
Relation Encoder
x_1 x_2 x_3 x_4 ... x_n
e_1 e_2
$(r_1 \ v_1 \ e_\alpha^1 \ e_\beta^1)$
$(r_2 \ v_2 \ e_\alpha^2 \ e_\beta^2)$
$(r_3 \ v_3 \ e_\alpha^3 \ e_\beta^3)$
$(r_4 \ v_4 \ e_\alpha^4 \ e_\beta^4)$
$(r_l \ v_l \ e_\alpha^l \ e_\beta^l)$
$(r_m \ v_m \ e_\alpha^m \ e_\beta^m)$
$(r_n \ v_n \ e_\alpha^n \ e_\beta^n)$
Class : r_i Coordinate: e_α, e_β Context : o_{v1}
Class : r_j Coordinate: e_β, e_θ Context : o_{v2}
Class : r_m Coordinate: e_θ, e_γ Context : o_{v3}
Class : r_n Coordinate: e_γ, e_δ Context : o_{v4}
Extraction of Global Level Relation
e_α R_t e_δ
Softmax Layer
Possible Global Relations are extracted from Constructed Knowledge Graph
Input Document
Split Document to Sentences
Local Relation Extraction for each Sentence
Convert Local Relations to Knowledge graph by Constructor

Figure 1. Overall process of GREG: Global level Relation Extraction with knowledge Graph embedding.

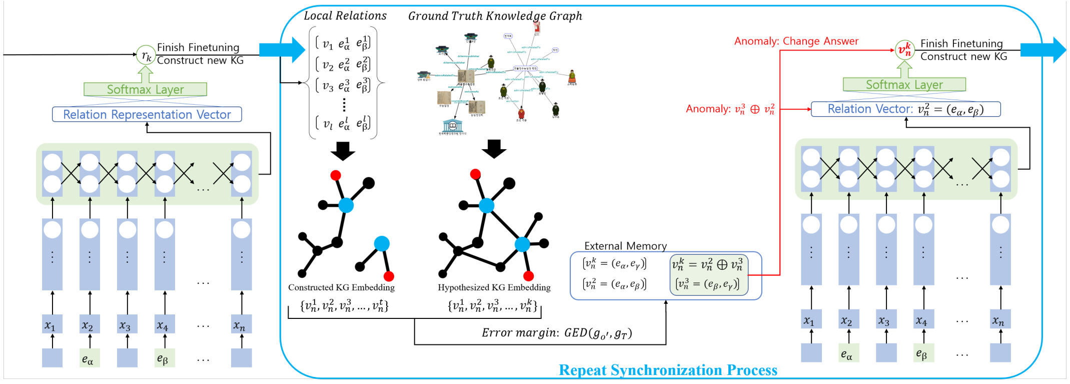
Finish Finetuning
Construct new KG
r_k
Softmax Layer
Relation Representation Vector
x_1 x_2 x_3 x_4 ... x_n
e_α e_β
Local Relations
Ground Truth Knowledge Graph
$(v_1 \ e_\alpha^1 \ e_\beta^1)$
$(v_2 \ e_\alpha^2 \ e_\beta^2)$
$(v_3 \ e_\alpha^3 \ e_\beta^3)$
$(v_l \ e_\alpha^l \ e_\beta^l)$
Constructed KG Embedding
$\{v_n^1, v_n^2, v_n^3, \ldots, v_n^t\}$
Hypothesized KG Embedding
$\{v_n^1, v_n^2, v_n^3, \ldots, v_n^k\}$
Error margin: $GED(g_{o'}, g_T)$
External Memory
$(v_n^k = (e_\alpha, e_\gamma))$
$(v_n^2 = (e_\alpha, e_\beta))$
$v_n^k = v_n^2 \oplus v_n^3$
$(v_n^3 = (e_\beta, e_\gamma))$
Anomaly: Change Answer
Anomaly: $v_n^3 \oplus v_n^2$
v_n^k
Finish Finetuning
Construct new KG
Softmax Layer
Relation Vector: $v_n^2 = (e_\alpha, e_\beta)$
x_1 x_2 x_3 x_4 ... x_n
e_α e_β
Repeat Synchronization Process

Figure 2. Singular Cycle of Synchronization process: Once the Local Relation Extraction module is trained, a full document is given as input sentence by sentence. The Local Relation info is then used to create a Knowledge Graph(KG). The constructed KG is then compared to a ground truth KG paired with the document. Both KGs are embedded based on the current LR representations. Any incorrect/missing relations within the constructed KG is stored in an external memory. As the fine-tuning of the local relation extraction module begins again, whenever a component from the external memory is detected, any paired vector representations are concatenated to it as well, allowing it to fine-tune against the incorrect/missing relations as well. The process aims to minimize the error margin between the two KGs that can be measured by graph edit distance.

Initially, the skeleton of the knowledge graph is constructed by using the sum of the local relation triplets. As common nodes between the triplets are used to link each entity, the skeleton knowledge graph contains all the local relation information from the given document input. Once the skeleton knowledge graph is complete, it is then converted into an embedding using the individual vector representations of local relations. Through these embeddings, we can construct new amalgam vector representations by combining the vectors that share entities. These amalgams are then sent to a common classifier used in the local relation extractor module—thus allowing us to identify possible global relations.

However, for this module to work, the first and second modules must be synchronized. Thus, every time a knowledge graph is constructed from a document, it is compared to a ground truth knowledge graph, creating an error margin. The synchronization process aims to readjust the knowledge graph embedder, relation classifier, and feature extractor to minimize the error margin. If the synchronization process is successful, the relations will be properly classified regardless of whether it was a local relation or global relation without any drops in performances.

3.1. Context-Aware Relation Extraction

The purpose of the first module is aimed to extract and classify local relations within a document. The overall module can be seen in Figure 3. In addition, to perform local relation extractions without suffering from a bottleneck issue [8], the input document is to be analyzed sentence by sentence. Once the document input is preprocessed into smaller sentences, the sentences are then given to a named entity recognition module to identify any existing entities within the sentence. In this research, we used two separate methods to extract named entities from the text for this experiment. The first module is the Stanford CoreNLP toolkit [18] developed by Manning et al. and the second named entity recognition module is a combined Bidirectional Long Short Term Memory and Convolutional Neural Network with a Conditional Random Field (Bi-LSTM-CNN-CRF) Korean named entity recognition module [17] developed by Kim et al. The reason behind using a Korean Named Entity recognition module is because the document-knowledge graph dataset used for the synchronization process of the two modules are made in Korean. Thus, to exclusively evaluate the model's viability on global relation extraction, the module has been trained to process both languages separately.

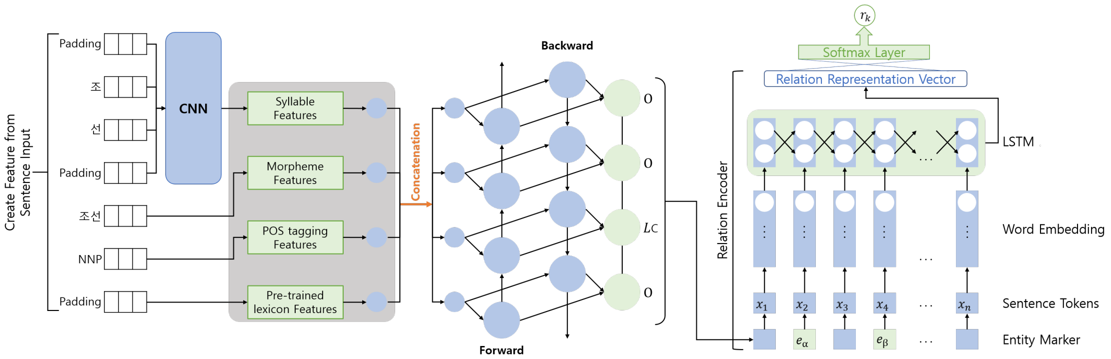

Figure 3. Long Short Term Memory(LSTM) based Mention-level 'Local' Relation Extraction model.

Given sentence x and its token as $x = \{x_1, x_2, x_3, ..., x_n\}$, each token within the sentence is to be embedded into a vector of k dimensions by using matrix $W \in R^{|V| \times k}$. Here, $|V|$ Indicates the size of the vocabulary from the word embedding. For this experiment, we used the 50-dimensional GloVe embeddings pre-trained on a 6 billion corpus by Pennington et al. [19] for our English module and a 200-dimensional Korean word embedding constructed from the Korean document dataset of history for our Korean module [6]. Additionally, the LSTM context-aware module has a layer size of 256 and the entity marker embeddings as $d = 3$. Once all the entities are recognized, every two entities from the same sentence are paired as candidate entities for a possible local relation. As the purpose of this module is to extract every inch of local relation information, every possible combination of two entities within a sentence is to be given input to the local relation extraction module.

For every pair of candidate entities given as inputs, along with the sentence it belongs to, the other entities within the sentence are temporarily considered as tokens. All tokens are then marked as to whether they belong to one of the candidate entities, or are independent based on their grammatical distances. These distances are measured through a part of speech tagging method [20]. A marker embedding matrix of $P \in R^{3 \times d}$ is randomly initialized of the three marker types: entity 1, entity 2, and none for independent tokens. Each token's marker embedding will then be concatenated to the word embedding with d as the dimension for the position embedding: (W_n, P_n).

By applying the sequence of n token embeddings to a recurrent neural network, the LSTM encoder converts it to a fixed-sized output vector $\boldsymbol{o_v}$. $\boldsymbol{o_v}$ acts as a representation of the relation between the target entities. The vector $\boldsymbol{o_v}$ is then passed to a softmax layer which will calculate which relation class it most likely belongs to. This can be shown as in Equation (1) with $P\,(r)$ as the classifier and the entities marked as e_1 and e_2, along with the softmax layer f_i defined in Equation (2):

$$P(r|\langle e_1, e_2\rangle, x; \theta) = \frac{\exp(f_r)}{\sum_{i=1}^{n_r}\exp(f_i)} \tag{1}$$

$$f_i = w_i \cdot o_v \oplus b_i \tag{2}$$

3.2. Knowledge Graph Constructor

However, the info extracted from the relation extraction module are only local relations limited to sentences. To expand our extraction to global relations, the second module receives the results from the first module. For each local relation, the two entities involved are considered as coordinates and are paired together with the relation's vector representation $\boldsymbol{o_v}$. These local relations and their representations are each used to construct the skeleton of a knowledge graph and its embedding.

As the initial knowledge graph is only a skeleton constructed from local relations, it does not contain any global relations as a ground truth knowledge graph would. Considering that the knowledge graph skeleton's embedding is composed of $O = \{v^1, v^2, v^3, \ldots, v^n\}$, we hypothesize that the true knowledge graph embedding should be composed of $T = \{t^1, t^2, t^3, \ldots, t^k\}$. The reason behind why we 'hypothesize' the knowledge graph is because, as we do not know what the values of the true knowledge graph's relation vector would be, we cannot get a ground truth embedding of it. Thus, the purpose of the second module is to 'recover missing' embedding components from T by using the skeleton knowledge graph's embedding O. These 'missing' embedding components are most likely global relations embeddings. In addition, the once recovered embeddings form knowledge graph $O' = \{v^1, v^2, v^3, \ldots, v^k\}$ and the newly founded embeddings in O' are used to extract global relations.

Since global relations can be broken down into multiple intermediate local relations. This can also mean that the sum of the intermediate relations shall indicate a global relation. For our ideal ground truth knowledge graph, its embedding should reflect this as well. Thus, the amalgamation of multiple intermediate local relation vectors should result in an accurate global relation vector within the knowledge graph's embedding. For example, if a fixed local relation vector with coordinate (e_1, e_2) is represented by v_a and fixed local relation vector with coordinate (e_2, e_3) is represented by v_b, then the global relation vector v_c of (e_1, e_3) must be represented as $\|v_c\| \approx \|v_a \oplus v_b\|$.

In addition, as each relation is represented as a vector, the vectors are required to represent global relations with the same consistency shown in local relation classification. Thus, if a global relation and a local relation are the same class, both vector representations must yield similar classification results. For example, let's say that the global relation of $\overrightarrow{AC}$ is "daughter" and local relation $\overrightarrow{XY}$ is also "daughter". In addition, the given intermediate local relation from $\overrightarrow{AC}$ is $\overrightarrow{AB}$ as "son" and $\overrightarrow{BC}$ as "sister". If consistency is to be met, the amalgam representation vectors of $\frac{\overrightarrow{AB} \oplus \overrightarrow{BC}}{2} = \overrightarrow{AC}$ should yield the same results as that of $\overrightarrow{XY}$ when it is classified by the relation classifier from the context-aware relation extraction module. In addition, as the global relation is classified, it becomes possible to convert the knowledge graph embedding back into a knowledge graph. Thus, the construction of a fully intact knowledge graph, including both local and global relations, of the document is finished.

However, during the early stages of training, this will most likely not be the case. Since amalgams of local relation vectors will not properly represent global relation when concatenated, to make this possible, as well as preserve the performance on local relation extraction, it needs to go through the 'Synchronization process'.

3.3. Synchronization Process

The synchronization process's main objective is to fine-tune the local relation extraction module's classifier by end-to-end fashion. This enables the classifier to properly classify both global relation embeddings and local relation vector representations consistently. The synchronization process largely consists of two parts: calculating error margin E for each document and applying feedback to the classifier. The error margin indicates the difference between the generated knowledge graph from the knowledge graph constructor and the ground truth knowledge graph from the training dataset.

To calculate the error margin, we used a toned-down graph edit distance [21] shown in Equation (3), where the cost of each graph edit operation $c(e_i)$ is always 1:

$$GED(g_{o'}, g_T) = \min_{(e_1, e_2, \ldots, e_k) \in P(g_{o'}, g_T)} \sum_{i=1}^{k} c(e) \tag{3}$$

In Equation (3), $g_{o'}$ indicates the generated knowledge graph while g_T is the ground truth knowledge graph. During this phase, any missing edges (entities) or vertex (relations) in the knowledge graphs are marked before the next phase begins. Usually, the missing entities from the generated knowledge graph $g_{o'}$ indicate two possibilities: that the entity was not detected by the NER module during the preprocessing stage, or that no relations were found connected to it. For the other relation errors, an external memory is given to store all the entities, local relation representation, and correct labels in pairs.

In the following feedback phase, the relation classifier from the first module starts fine-tuning. Then, the fine-tuning will take place to increase the performance of the classifier. However, to synchronize the classifier's fine-tuning while training the knowledge graph embeddings from our Knowledge graph module, we added an anomaly system. When an anomaly occurs, the external memory is checked to see whether errors involving the current input exist. If an error with the input class involved is detected, then any other involving relation vector representations that were paired to it are immediately concatenated to the current input's relation vector representation—thus allowing the model not only to fine-tune itself but correct the errors found during the previous phase. The re-occurrence of whether an anomaly happens is based on each error margin. The higher the error margin was for each error, the higher it will reoccur during fine-tuning. Once the fine-tuning process is done, then we return to phase 1 and repeat until the error margins are minimized. For this process, the individual fine-tuning process of phase 2 was always limited to 200 epochs to avoid any possible overfitting—allowing the constructed knowledge graph to become close to an ideal knowledge graph state similar to the ground truth knowledge graph as shown in Figure 4.

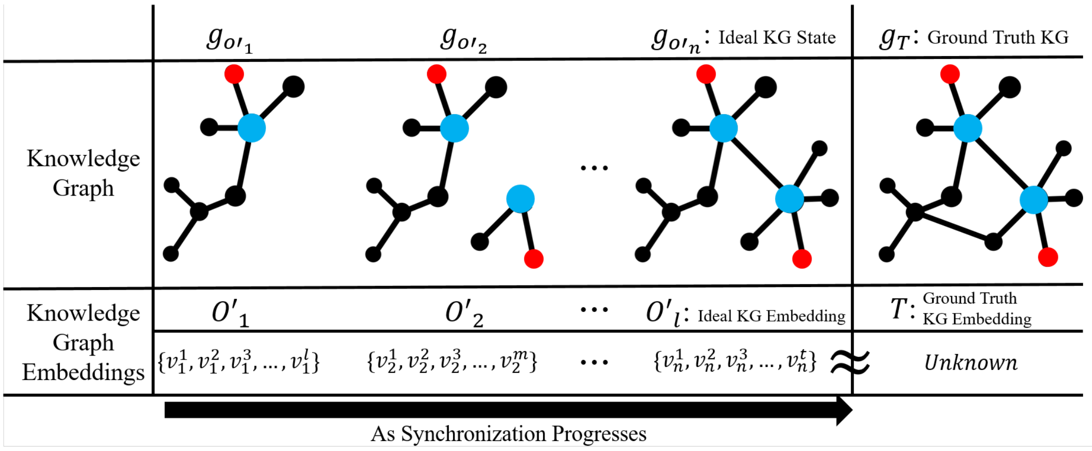
$g_{o'_1}$
$g_{o'_2}$
$g_{o'_n}$: Ideal KG State
g_T: Ground Truth KG
Knowledge Graph
...
Knowledge Graph Embeddings
O'_1
O'_2
...
O'_l: Ideal KG Embedding
T: Ground Truth KG Embedding
$\{v_1^1, v_1^2, v_1^3, \ldots, v_1^l\}$
$\{v_2^1, v_2^2, v_2^3, \ldots, v_2^m\}$
...
$\{v_n^1, v_n^2, v_n^3, \ldots, v_n^t\}$
$\approx$
Unknown
As Synchronization Progresses

Figure 4. Progression of knowledge graph output as the synchronization process takes place. As the synchronization process progresses, one can see the KGs slowly start to resemble the Ground Truth KG. As it is impossible to know what the ground truth KGs true embedding is, we hypothesize that the ideal constructed KG will have similar embeddings to that of the ground truth KG.

4. Experiment and Results

Several training datasets were used for the training of this model: the Wikipedia-Wikidata sentence-level relation annotations by Sorokin et al. [16], the DocRED dataset from Yuan et al. [4], and the Korean document dataset of history [6]. The purpose of the presented experiments is as follows: to compare the performances of the LSTM relation extraction module before and after the synchronization process, and compare our proposed model's overall relation extraction performances to other models.

4.1. Effects of Synchronization

Our first experiment attempts to confirm whether the synchronization process does or does not hinder the local relation extraction module's performance. Thus, we have measured the module's performance on different periods during the synchronization process. In Table 1, one can observe that once the synchronization process begins the performance immediately dropped drastically. This is most likely due to the fact that, during the initial stages during synchronization, the process alters the local relation extraction module with no regard to the module's original purpose. In addition, notice that, despite the Korean dataset [6] being smaller compared to the DocRED dataset [4], it requires more time as well as shows a much steeper performance drop in the initial stages of the training. However, eventually we can observe both incidents returning to their original performance levels before the module starts overfitting. This indicates that the synchronization process does hinder the module's performance in its early stages. However, as the process progresses, the defection towards the local relation extraction module lessens to a point that it becomes negligible.

Table 1. Performance in the LSTM based Relation Extraction module during the process of alternated training.

	Initial Training Only	**Alternated Training**					
		1st Epoch	**10th Epoch**	**50th Epoch**	**100th Epoch**	**200th Epoch**	**500th Epoch**
Korean dataset	95.03%	67.60%	70.78%	84.78%	88.99%	93.64%	93.61%
DocRED dataset	47.83%	29.53%	33.33%	35.55%	42.02%	51.07%	51.05%

4.2. Model Performance Comparison

Our final experiment is comparing our proposed model's performance to other existing models. For this, we compared the proposed model's performance from three different perspectives. For this experiment, we will measure the module's capabilities on local relation extraction, global relation extraction, and overall performance. The following models shall be used for comparison: an End-to-End Convolutional Neural Network(CNN) based model by Nguyen et al. [22] and the End-to-End LSTM based model by Miwa et al. [23] shall act as our baseline comparisons. This will be followed by the context-aware LSTM based relation extraction module by Sorokin et al. [16], which forms the base for the first part of our module. Additionally, to evaluate our proposed approach itself, we have also added two variations of our model that use the End-to-End CNN based model [22] and LSTM based model [23] to act as the first part of our model. Each of the three tasks is tested through different datasets. The Wikipedia-Wikidata sentence-level relation annotations [16] shall be used for the local relation extraction task, and the Korean document dataset of history [6] shall be used for the global relation extraction task. For our measurement of overall performance, we used both the Korean document dataset of history [6] and DocRED dataset [4] and evaluated their performances separately. Among the 10,470 lines of documented Korean text of history [6] used for this task, we separated 1201 lines to be used as a test set, and 550 lines to be used as a validation set. Thus, each of these 1751 lines of text was not used as a training set. This is approximately about 7.5% and 3% of the whole dataset. The total sum of the two consists of 50 documents and corresponding knowledge graphs out of the entire 503 documents. On the other hand, the DocRED dataset contains 132,375 entities and 56,354 relational facts annotated on 5053 Wikipedia documents. Additionally, as at least 40.7% of the relational facts in DocRED are multisentence level relations, it is a perfect dataset to evaluate the overall performance of this model. Both datasets are human-annotated document-level datasets.

As seen in Table 2, the LSTM based models tend to exceed the CNN based models. The main difference seems to come from the fact that the CNN based models tend to perform worse than the other LSTM based models in general.

Table 2. Performance comparisons on three different levels: Local relations, Global relations, Overall relations by the use of Precision(P), Recall(R) and F1-score(F1) (%)

		CNN	LSTM	LSTM Context	GREG var. (CNN)	GREG var. (LSTM)	GREG (Context)
Wikidata Sentence data (Local)	**P**	71.25	75.79	79.62	71.54	74.71	79.03
	R	53.91	70.78	80.75	52.83	69.08	78.83
	F1	61.37	73.20	80.18	60.78	71.78	78.93
Korean Document data (Global)	**P**	19.40	18.33	38.30	30.48	33.78	51.32
	R	31.31	33.19	28.72	44.33	48.34	46.14
	F1	23.95	23.61	32.82	36.12	39.77	48.59
Korean Document data (Overall)	**P**	47.98	52.11	53.46	47.73	53.43	54.74
	R	40.32	41.31	63.55	40.38	44.91	66.85
	F1	43.82	46.09	58.07	43.75	48.40	60.19
DocRED data (Overall)	**P**	50.29	54.69	50.31	50.53	52.29	51.43
	R	35.54	47.22	51.89	37.94	46.29	54.41
	F1	42.33	50.68	51.09	43.34	49.11	52.88

First, we compare the models' performances on local relation extraction. Initially, one can see that the CNN baseline model [22] performs the worst with the LSTM baseline model [23] performing almost the same. The LSTM Context model [16] is the highest performer with the GREG(Context) following behind. The performance difference between the LSTM Context model [16] and GREG(Context) on the local extraction task once again indicates the facts we found in our previous experiment. In addition, we can suspect this is a universal effect as the performance defection from the synchronization process seems to apply to the other GREG variants.

However, this changes drastically when we go to the second task: comparing global extractions. In this task, each input is given as documents, and only relations that are labeled as 'global relations' within the test set are evaluated. Here, we can see our proposed model truly shining. All three GREG variants show a significantly higher F1 score compared to the other models. Additionally, one can observe that the GREG variants reflect the performances of their local relation extraction module, indicating that our proposed approach's performance is largely affected by how its local relation extraction module performs on local relation extraction. Thus, we can observe the GREG(Context) showing the highest performance, followed by variants GREG(LSTM) and GREG(CNN) accordingly, which matches how the LSTM Context model [16], LSTM baseline model [23], and CNN baseline model [22] perform. For the baseline models, their low performances are likely caused by the bottleneck problem [8] triggered by the document sized input.

Finally, we see how each model fares in overall performance. While we used the standard test set for the DocRED dataset [4] to evaluate the models' ability to extract relations in various lengths of input, we used the Korean Documents of History dataset [6] to measure each model's overall performance on dealing with document inputs in general. Therefore, when using our Korean dataset, each model is given a document sized inputs for evaluation. However, while the GREG variants are only given the document as a single input, the baseline models received the input by both document and sentence by sentence. This enables a fair overall performance comparison on both local and global relation extractions. As seen in Table 2, all three GREG models vastly outperform the models they are based off. Proving the proposed method greatly advances the model's performance in dealing with documents. By pure performance, we can observe our original model GREG(Context) showing the highest performance, followed by the LSTM Context Model [16] and the GREG(LSTM) variant. We suspect that, despite the latter GREG variants displaying higher performances in global relation extraction, the other two GREG variants came up short because the number of local relations vastly outnumber the number of global relations within each document—thus leading to an overall lower F1-score in general. This again strengthens our suspicion that our proposed method depends greatly on which local relation extraction model is used.

Through these evaluations, we presented that our proposed module shows the highest performance among all others in global relation extraction. Even when we evaluate overall performance, it still performs better than all the other models by a significant amount. Proving our proposed approach is an effective way to extract global relations from documents while minimizing the loss of performance on local relation extractions. The advantage we have over the other models is simple. Other models struggle to contain and compress as much data as they can, yet attempting to not lose any vital information during a single process. Our approach allows the model not to suffer from this problem by taking a divide and conquer approach. Extracting each vital information from a sentence one at a time enables to not lose any vital information. In addition, when it recreates the gathered summary into a knowledge graph, it enables the model to recognize all the vital information it needs for global relation extraction—thus allowing the task to be done without having to bite more than the model can chew.

5. Conclusions

In a time where data are increasing at an enormous rate, it has become impossible to convert unstructured text into structured data by human hands. In addition, as the size of unstructured data along with their number increases, the importance of automating this task is one that will only increase its value as time passes.

For this article, we have presented a Global level Relation Extraction model with knowledge graph embedding for document-level inputs. When converting relations within unstructured text to structured data, we focused on the fact that relations described throughout the document tend to represent the most vital information within the document [24]. Thus, there have been multiple attempts to extract such global relations from documents or large bodies of text. However, the re-occurring problem in this task is the bottleneck problem [8]. Because most models attempt to process the entire document at once, the model suffers from loss of information. To overcome this dilemma, we have proposed a divide and conquer approach to avoid this problem entirely by using two separate models to extract local relations and construct a knowledge graph using the local relations. The knowledge graph is then embedded into a vector space and used to predict possible global relations within the document. Thus, for this method to work, we applied a synchronization process to the two modules so that the constructed knowledge graphs embedding would properly represent the global relation as well as perform decent local relation extraction.

For our future work, we plan to ease the pressure that comes from inquiring training data. Even for this task, the pressure from not having sufficient data was immensely heavy, especially when dealing with large bodies of text. Therefore, we will try to attempt an unsupervised version of our method that requires little to no data, which we hope will allow us to come closer to fully automating this task in the near future.

Author Contributions: Conceptualization, formal analysis, methodology, software, visualization, writing—original draft preparation, and writing—review and editing, K.K.; investigation, data curation, and resources, K.K., Y.H., and G.K.; validation, supervision, project administration, and funding acquisition, H.L. All authors have read and agree to the published version of the manuscript.

Acknowledgments: This research is supported by the Ministry of Culture, Sport, and Tourism (MCST) and the Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research and Development Program 2017 (No. R2017030045).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Young, T.; Hazarika, D.; Poria, S.; Cambria, E. Recent trends in deep learning based natural language processing. *IEEE Comput. Intell. Mag.* **2018**, *13*, 55–75. [CrossRef]
2. Blumberg, R.; Atre, S. The problem with unstructured data. *DM Rev.* **2003**, *13*, 62.
3. Peng, N.; Poon, H.; Quirk, C.; Toutanova, K.; Yih, W.T. Cross-sentence n-ary relation extraction with graph lstms. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 101–115. [CrossRef]
4. Yao, Y.; Ye, D.; Li, P.; Han, X.; Lin, Y.; Liu, Z.; Liu, Z.; Huang, L.; Zhou, J.; Sun, M. DocRED: A Large-Scale Document-Level Relation Extraction Dataset. In Proceedings of the ACL 2019, Austin, TX, USA, 4–13 October 2019.
5. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
6. Kim, H. Digital Humanities—Research Administration of Archives and Humanities. 2017. Available online: http://dh.aks.ac.kr/Encyves/wiki/ (accessed on 11 October 2019).
7. Moschitti, A.; Patwardhan, S.; Welty, C. Long-distance time-event relation extraction. In Proceedings of the Sixth International Joint Conference on Natural Language Processing, Nagoya, Japan, 14–18 October 2013; pp. 1330–1338.
8. Tishby, N.; Zaslavsky, N. Deep learning and the information bottleneck principle. In Proceedings of the 2015 IEEE Information Theory Workshop (ITW), Jerusalem, Israel, 26 April–1 May 2015; pp. 1–5.
9. Csáji, B.C. *Approximation with Artificial Neural Networks*; Faculty of Sciences, Etvs Lornd University: Budapest, Hungary, 2001; Volume 24, p. 48.
10. Gerber, M.; Chai, J. Beyond NomBank: A Study of Implicit Arguments for Nominal Predicates. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*; Association for Computational Linguistics: Uppsala, Sweden, 2010; pp. 1583–1592.
11. Quirk, C.; Poon, H. Distant Supervision for Relation Extraction beyond the Sentence Boundary. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*; Association for Computational Linguistics: Valencia, Spain, 2017; pp. 1171–1182.
12. Cimiano, P. *Ontologies*; Springer: New York, NY, USA, 2006.
13. Wang, Q.; Mao, Z.; Wang, B.; Guo, L. Knowledge graph embedding: A survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2724–2743. [CrossRef]
14. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge graph embedding by translating on hyperplanes. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, Quebec City, QC, Canada, 27–31 July 2014.
15. Lu, D.; Whitehead, S.; Huang, L.; Ji, H.; Chang, S.F. Entity-aware Image Caption Generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Brussels, Belgium, 2018; pp. 4013–4023. [CrossRef]
16. Sorokin, D.; Gurevych, I. Context-aware representations for knowledge base relation extraction. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 1784–1789.
17. Kim, G.; Kim, K.; Jo, J.; Lim, H. Constructing for Korean Traditional culture Corpus and Development of Named Entity Recognition Model using Bi-LSTM-CNN-CRFs. *J. Korea Converg. Soc.* **2018**, *9*, 47–52.
18. Manning, C.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S.; McClosky, D. The Stanford CoreNLP natural language processing toolkit. In Proceedings of the 52nd Annual Meeting of the Association For Computational Linguistics: System Demonstrations, Baltimore, MD, USA, 23–24 June 2014; pp. 55–60.
19. Pennington, J.; Socher, R.; Manning, C. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
20. Park, E.L.; Cho, S. KoNLPy: Korean natural language processing in Python. In Proceedings of the 26th Annual Conference on Human & Cognitive Language Technology, Beijing, China, 1–2 November 2014; Volume 6.
21. Sanfeliu, A.; Fu, K.S. A distance measure between attributed relational graphs for pattern recognition. *IEEE Trans. Syst. Man Cybern.* **1983**, pp. 353–362. [CrossRef]

22. Nguyen, T.H.; Grishman, R. Relation extraction: Perspective from convolutional neural networks. In Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, Denver, CO, USA, 5 June 2015; pp. 39–48.
23. Miwa, M.; Bansal, M. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; Association for Computational Linguistics: Berlin, Germany, 2016; pp. 1105–1116. [CrossRef]
24. Konstantinova, N. Review of relation extraction methods: What is new out there? In *Proceedings of the International Conference on Analysis of Images, Social Networks and Texts*; Springer: Cham, Switzerland, 2014; pp. 15–28.