Neural Relation Extraction for Knowledge Base Enrichment

Bayu Distiawan Trisedya[1], **Gerhard Weikum**[2], **Jianzhong Qi**[1], **Rui Zhang**[1*]
[1] The University of Melbourne, Australia
[2] Max Planck Institute for Informatics, Saarland Informatics Campus, Germany
{btrisedya@student, jianzhong.qi@, rui.zhang@}unimelb.edu.au
weikum@mpi-inf.mpg.de

Abstract

We study relation extraction for knowledge base (KB) enrichment. Specifically, we aim to extract entities and their relationships from sentences in the form of triples and map the elements of the extracted triples to an existing KB in an end-to-end manner. Previous studies focus on the extraction itself and rely on Named Entity Disambiguation (NED) to map triples into the KB space. This way, NED errors may cause extraction errors that affect the overall precision and recall. To address this problem, we propose an end-to-end relation extraction model for KB enrichment based on a neural encoder-decoder model. We collect high-quality training data by distant supervision with co-reference resolution and paraphrase detection. We propose an n-gram based attention model that captures multi-word entity names in a sentence. Our model employs jointly learned word and entity embeddings to support named entity disambiguation. Finally, our model uses a modified beam search and a triple classifier to help generate high-quality triples. Our model outperforms state-of-the-art baselines by 15.51% and 8.38% in terms of F1 score on two real-world datasets.

1 Introduction

Knowledge bases (KBs), often in the form of knowledge graphs (KGs), have become essential resources in many tasks including Q&A systems, recommender system, and natural language generation. Large KBs such as DBpedia (Auer et al., 2007), Wikidata (Vrandecic and Krötzsch, 2014) and Yago (Suchanek et al., 2007) contain millions of facts about entities, which are represented in the form of subject-predicate-object triples. However, these KBs are far from complete and mandate continuous enrichment and curation.

*Rui Zhang is the corresponding author.

Input sentence:
"New York University is a private university in Manhattan."
Unsupervised approach output:
⟨NYU, is, private university⟩
⟨NYU, is private university in, Manhattan⟩
Supervised approach output:
⟨NYU, instance of, Private University⟩
⟨NYU, located in, Manhattan⟩
Canonicalized output:
⟨Q49210, P31, Q902104⟩
⟨Q49210, P131, Q11299⟩

Table 1: Relation extraction example.

Previous studies work on embedding-based model (Nguyen et al., 2018; Wang et al., 2015) and entity alignment model (Chen et al., 2017; Sun et al., 2017; Trisedya et al., 2019) to enrich a knowledge base. Following the success of the sequence-to-sequence architecture (Bahdanau et al., 2015) for generating sentences from structured data (Marcheggiani and Perez-Beltrachini, 2018; Trisedya et al., 2018), we employ this architecture to do the opposite, which is extracting triples from a sentence.

In this paper, we study how to enrich a KB by relation extraction from textual sources. Specifically, we aim to extract triples in the form of $\langle h, r, t \rangle$, where h is a head entity, t is a tail entity, and r is a relationship between the entities. Importantly, as KBs typically have much better coverage on entities than on relationships, we assume that h and t are existing entities in a KB, r is a predicate that falls in a predefined set of predicates we are interested in, but the relationship $\langle h, r, t \rangle$ does not exist in the KB yet. We aim to find more relationships between h and t and add them to the KB. For example, from the first extracted triples in Table 1 we may recognize two entities "NYU" (abbreviation of New York University) and "Private University", which already exist in the KB;

also the predicate "instance of" is in the set of predefined predicates we are interested in, but the relationship of ⟨NYU, instance of, Private University⟩ does not exist in the KB. We aim to add this relationship to our KB. This is the typical situation for KB enrichment (as opposed to constructing a KB from scratch or performing relation extraction for other purposes, such as Q&A or summarization).

KB enrichment mandates that the entities and relationships of the extracted triples are canonicalized by mapping them to their proper entity and predicate IDs in a KB. Table 1 illustrates an example of triples extracted from a sentence. The entities and predicate of the first extracted triple, including NYU, instance of, and Private University, are mapped to their unique IDs Q49210, P31, and Q902104, respectively, to comply with the semantic space of the KB.

Previous studies on relation extraction have employed both unsupervised and supervised approaches. Unsupervised approaches typically start with a small set of manually defined extraction patterns to detect entity names and phrases about relationships in an input text. This paradigm is known as *Open Information Extraction* (Open IE) (Banko et al., 2007; Corro and Gemulla, 2013; Gashteovski et al., 2017). In this line of approaches, both entities and predicates are captured in their surface forms without canonicalization. Supervised approaches train statistical and neural models for inferring the relationship between two known entities in a sentence (Mintz et al., 2009; Riedel et al., 2010, 2013; Zeng et al., 2015; Lin et al., 2016). Most of these studies employ a preprocessing step to recognize the entities. Only few studies have fully integrated the mapping of extracted triples onto uniquely identified KB entities by using logical reasoning on the existing KB to disambiguate the extracted entities (e.g., (Suchanek et al., 2009; Sa et al., 2017)).

Most existing methods thus entail the need for *Named Entity Disambiguation* (NED) (cf. the survey by Shen et al. (2015)) as a separate processing step. In addition, the mapping of relationship phrases onto KB predicates necessitates another mapping step, typically aided by paraphrase dictionaries. This two-stage architecture is inherently prone to error propagation across its two stages: NED errors may cause extraction errors (and vice versa) that lead to inaccurate relationships being added to the KB.

We aim to integrate the extraction and the canonicalization tasks by proposing an end-to-end neural learning model to jointly extract triples from sentences and map them into an existing KB. Our method is based on the encoder-decoder framework (Cho et al., 2014) by treating the task as a translation of a sentence into a sequence of elements of triples. For the example in Table 1, our model aims to translate "New York University is a private university in Manhattan" into a sequence of IDs "Q49210 P31 Q902104 Q49210 P131 Q11299", from which we can derive two triples to be added to the KB.

A standard encoder-decoder model with attention (Bahdanau et al., 2015) is, however, unable to capture the multi-word entity names and verbal or noun phrases that denote predicates. To address this problem, we propose a novel form of n-gram based attention that computes the n-gram combination of attention weight to capture the verbal or noun phrase context that complements the word level attention of the standard attention model. Our model thus can better capture the multi-word context of entities and relationships. Our model harnesses pre-trained word and entity embeddings that are jointly learned with skip gram (Mikolov et al., 2013) and TransE (Bordes et al., 2013). The advantages of our jointly learned embeddings are twofold. First, the embeddings capture the relationship between words and entities, which is essential for named entity disambiguation. Second, the entity embeddings preserve the relationships between entities, which help to build a highly accurate classifier to filter the invalid extracted triples. To cope with the lack of fully labeled training data, we adapt distant supervision to generate aligned pairs of sentence and triple as the training data. We augment the process with co-reference resolution (Clark and Manning, 2016) and dictionary-based paraphrase detection (Ganitkevitch et al., 2013; Grycner and Weikum, 2016). The co-reference resolution helps extract sentences with implicit entity names, which enlarges the set of candidate sentences to be aligned with existing triples in a KB. The paraphrase detection helps filter sentences that do not express any relationships between entities.

The main contributions of this paper are:

- We propose an end-to-end model for extract-

ing and canonicalizing triples to enrich a KB. The model reduces error propagation between relation extraction and NED, which existing approaches are prone to.

- We propose an n-gram based attention model to effectively map the multi-word mentions of entities and their relationships into uniquely identified entities and predicates. We propose joint learning of word and entity embeddings to capture the relationship between words and entities for named entity disambiguation. We further propose a modified beam search and a triple classifier to generate high-quality triples.
- We evaluate the proposed model over two real-world datasets. We adapt distant supervision with co-reference resolution and paraphrase detection to obtain high-quality training data. The experimental results show that our model consistently outperforms a strong baseline for neural relation extraction (Lin et al., 2016) coupled with state-of-the-art NED models (Hoffart et al., 2011; Kolitsas et al., 2018).

2 Related Work

2.1 Open Information Extraction

Banko et al. (2007) introduced the paradigm of Open Information Extraction (Open IE) and proposed a pipeline that consists of three stages: learner, extractor, and assessor. The learner uses dependency-parsing information to learn patterns for extraction, in an unsupervised way. The extractor generates candidate triples by identifying noun phrases as arguments and connecting phrases as predicates. The assessor assigns a probability to each candidate triple based on statistical evidence. This approach was prone to extracting incorrect, verbose and uninformative triples. Various follow-up studies (Fader et al., 2011; Mausam et al., 2012; Angeli et al., 2015; Mausam, 2016) improved the accuracy of Open IE, by adding handcrafted patterns or by using distant supervision. Corro and Gemulla (2013) developed ClausIE, a method that analyzes the clauses in a sentence and derives triples from this structure. Gashteovski et al. (2017) developed MinIE to advance ClausIE by making the resulting triples more concise.

Stanovsky et al. (2018) proposed a supervised learner for Open IE by casting relation extraction into sequence tagging. A bi-LSTM model is trained to predict the label (entity, predicate, or other) of each token of the input. The work most related to ours is Neural Open IE (Cui et al., 2018), which proposed an encoder-decoder with attention model to extract triples. However, this work is not geared for extracting relations of canonicalized entities. Another line of studies use neural learning for semantic role labeling (He et al., 2018), but the goal here is to recognize the predicate-argument structure of a single input sentence – as opposed to extracting relations from a corpus.

All of these methods generate triples where the head and tail entities and the predicate stay in their surface forms. Therefore, different names and phrases for the same entities result in multiple triples, which would pollute the KG if added this way. The only means to map triples to uniquely identified entities in a KG is by post-processing via entity linking (NED) methods (Shen et al., 2015) or by clustering with subsequent mapping (Galárraga et al., 2014).

2.2 Entity-aware Relation Extraction

Inspired by the work of Brin (1998), state-of-the-art methods employ distant supervision by leveraging seed facts from an existing KG (Mintz et al., 2009; Suchanek et al., 2009; Carlson et al., 2010). These methods learn extraction patterns from seed facts, apply the patterns to extract new fact candidates, iterate this principle, and finally use statistical inference (e.g., a classifier) for reducing the false positive rate. Some of these methods hinge on the assumption that the co-occurrence of a seed fact's entities in the same sentence is an indicator of expressing a semantic relationship between the entities. This is a potential source of wrong labeling. Follow-up studies (Hoffmann et al., 2010; Riedel et al., 2010, 2013; Surdeanu et al., 2012) overcome this limitation by various means, including the use of relation-specific lexicons and latent factor models. Still, these methods treat entities by their surface forms and disregard their mapping to existing entities in the KG.

Suchanek et al. (2009) and Sa et al. (2017) used probabilistic-logical inference to eliminate false positives, based on constraint solving or Monte Carlo sampling over probabilistic graphical models, respectively. These methods integrate entity linking (i.e., NED) into their models. However, both have high computational complexity and rely on modeling constraints and appropriate priors.

Recent studies employ neural networks to learn the extraction of triples. Nguyen and Grish-

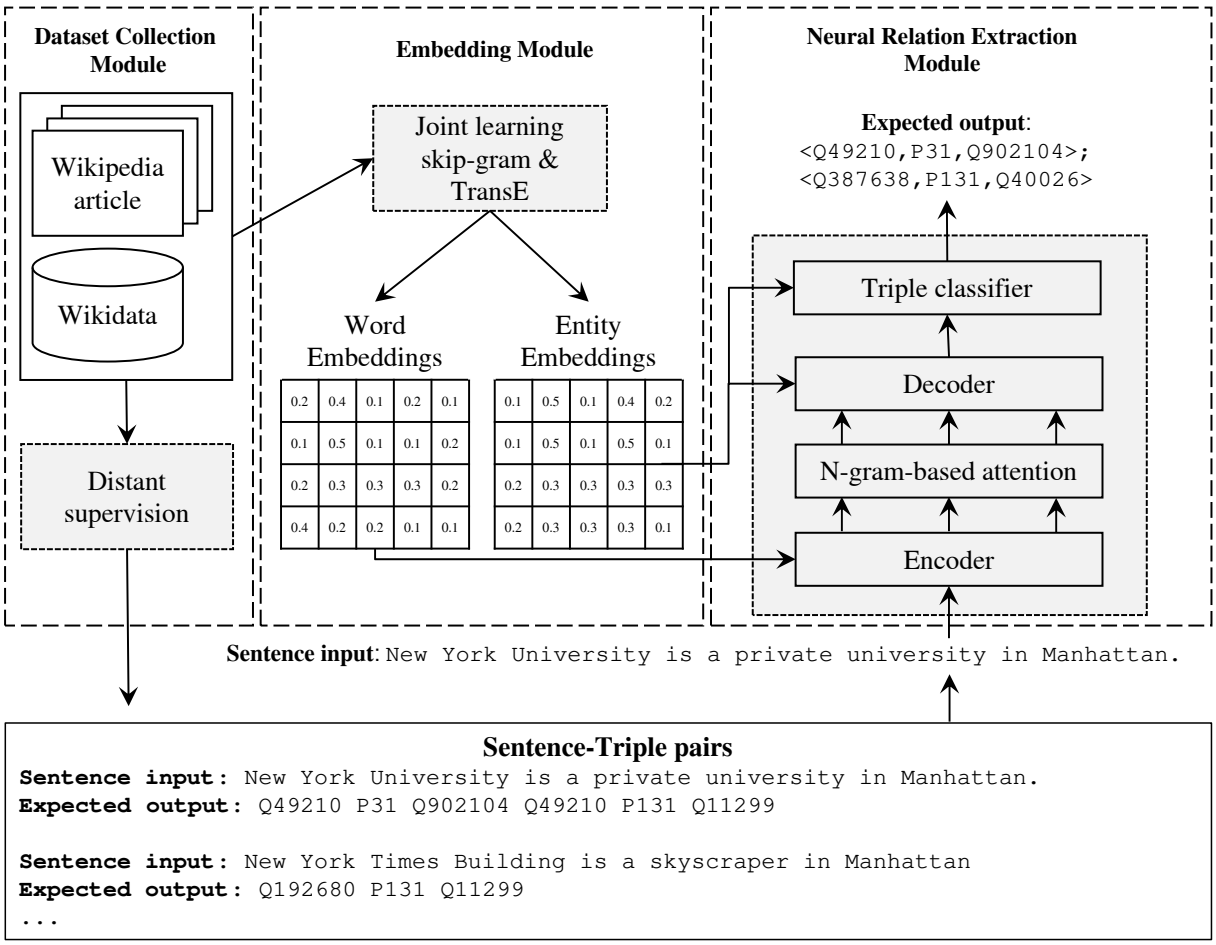

Figure 1: Overview of our proposed solution.

man (2015) proposed Convolution Networks with multi-sized window kernel. Zeng et al. (2015) proposed Piecewise Convolution Neural Networks (PCNN). Lin et al. (2016, 2017) improved this approach by proposing PCNN with sentence-level attention. This method performed best in experimental studies; hence we choose it as the main baseline against which we compare our approach. Follow-up studies considered further variations: Zhou et al. (2018) proposed hierarchical attention, Ji et al. (2017) incorporated entity descriptions, Miwa and Bansal (2016) incorporated syntactic features, and Sorokin and Gurevych (2017) used background knowledge for contextualization.

None of these neural models is geared for KG enrichment, as the canonicalization of entities is out of their scope.

3 Proposed Model

We start with the problem definition. Let $G = (E, R)$ be an existing KG where E and R are the sets of entities and relationships (predicates) in G, respectively. We consider a sentence $S = \langle w_1, w_2, ..., w_i \rangle$ as the input, where w_i is a token at position i in the sentence. We aim to extract a set of triples $O = \{o_1, o_2, ..., o_j\}$ from the sentence, where $o_j = \langle h_j, r_j, t_j \rangle$, $h_j, t_j \in E$, and $r_j \in R$. Table 1 illustrates the input and target output of our problem.

3.1 Solution Framework

Figure 1 illustrates the overall solution framework. Our framework consists of three components: *data collection* module, *embedding* module, and *neural relation extraction* module.

In the data collection module (detailed in Section 3.2), we align known triples in an existing KB with sentences that contain such triples from a text corpus. The aligned pairs of sentences and triples will later be used as the training data in our neural relation extraction module. This alignment is done by distant supervision. To obtain a large number of high-quality alignments, we augment the process with co-reference resolution to extract sentences with implicit entity names, which enlarges the set of candidate sentences to be aligned. We further

use dictionary based paraphrase detection to filter sentences that do not express any relationships between entities.

In the embedding module (detailed in Section 3.3), we propose a joint learning of word and entity embeddings by combining skip-gram (Mikolov et al., 2013) to compute the word embeddings and TransE (Bordes et al., 2013) to compute the entity embeddings. The objective of the joint learning is to capture the similarity of words and entities that helps map the entity names into the related entity IDs. Moreover, the resulting entity embeddings are used to train a triple classifier that helps filter invalid triples generated by our neural relation extraction model.

In the neural relation extraction module (detailed in Section 3.4), we propose an n-gram based attention model by expanding the attention mechanism to the n-gram token of a sentence. The n-gram attention computes the n-gram combination of attention weight to capture the verbal or noun phrase context that complements the word level attention of the standard attention model. This expansion helps our model to better capture the multi-word context of entities and relationships.

The output of the encoder-decoder model is a sequence of the entity and predicate IDs where every three IDs indicate a triple. To generate high-quality triples, we propose two strategies. The first strategy uses a modified beam search that computes the lexical similarity of the extracted entities with the surface form of entity names in the input sentence to ensure the correct entity prediction. The second strategy uses a triple classifier that is trained using the entity embeddings from the joint learning to filter the invalid triples. The triple generation process is detailed in Section 3.5

3.2 Dataset Collection

We aim to extract triples from a sentence for KB enrichment by proposing a supervised relation extraction model. To train such a model, we need a large volume of fully labeled training data in the form of sentence-triple pairs. Following Sorokin and Gurevych (2017), we use distant supervision (Mintz et al., 2009) to align sentences in Wikipedia[1] with triples in Wikidata[2] (Vrandecic and Krötzsch, 2014).

[1]https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2

[2]https://dumps.wikimedia.org/wikidatawiki/entities/latest-all.ttl.gz

We map an entity mention in a sentence to the corresponding entity entry (i.e., Wikidata ID) in Wikidata via the hyperlink associated to the entity mention, which is recorded in Wikidata as the `url` property of the entity entry. Each pair may contain one sentence and multiple triples. We sort the order of the triples based on the order of the predicate paraphrases that indicate the relationships between entities in the sentence. We collect sentence-triple pairs by extracting sentences that contain both head and tail entities of Wikidata triples. To generate high-quality sentence-triple pairs, we propose two additional steps: (1) extracting sentences that contain implicit entity names using co-reference resolution, and (2) filtering sentences that do not express any relationships using paraphrase detection. We detail these steps below.

Prior to aligning the sentences with triples, in Step (1), we find the implicit entity names to increase the number of candidate sentences to be aligned. We apply co-reference resolution (Clark and Manning, 2016) to each paragraph in a Wikipedia article and replace the extracted co-references with the proper entity name. We observe that the first sentence of a paragraph in a Wikipedia article may contain a pronoun that refers to the main entity. For example, there is a paragraph in the `Barack Obama` article that starts with a sentence `"He was reelected to the Illinois Senate in 1998"`. This may cause the standard co-reference resolution to miss the implicit entity names for the rest of the paragraph. To address this problem, we heuristically replace the pronouns in the first sentence of a paragraph if the main entity name of the Wikipedia page is not mentioned. For the sentence in the previous example, we replace `"He"` with `"Barack Obama"`. The intuition is that a Wikipedia article contains content of a single entity of interest, and that the pronouns mentioned in the first sentence of a paragraph mostly relate to the main entity.

In Step (2), we use dictionary based paraphrase detection to capture relationships between entities in a sentence. First, we create a dictionary by populating predicate paraphrases from three sources including PATTY (Nakashole et al., 2012), POLY (Grycner and Weikum, 2016), and PPDB (Ganitkevitch et al., 2013) that yield 540 predicates and 24,013 unique paraphrases. For example, predicate paraphrases for the relation-

	#pairs	#triples	#entities	#predicates
All (WIKI)	255,654	330,005	279,888	158
Train+val	225,869	291,352	249,272	157
Test (WIKI)	29,785	38,653	38,690	109
Test (GEO)	1,000	1,095	124	11

Table 2: Statistics of the dataset.

ship "place of birth" are {born in, was born in, ...}. Then, we use this dictionary to filter sentences that do not express any relationships between entities. We use exact string matching to find verbal or noun phrases in a sentence which is a paraphrases of a predicate of a triple. For example, for the triple ⟨Barack Obama, place of birth, Honolulu⟩, the sentence "Barack Obama was born in 1961 in Honolulu, Hawaii" will be retained while the sentence "Barack Obama visited Honolulu in 2010" will be removed (the sentence may be retained if there is another valid triple ⟨Barack Obama, visited, Honolulu⟩). This helps filter noises for the sentence-triple alignment.

The collected dataset contains 255,654 sentence-triple pairs. For each pair, the maximum number of triples is four (i.e., a sentence can produce at most four triples). We split the dataset into train set (80%), dev set (10%) and test set (10%) (we call it the **WIKI** test dataset). For *stress testing* (to test the proposed model on a different style of text than the training data), we also collect another test dataset outside Wikipedia. We apply the same procedure to the user reviews of a travel website. First, we collect user reviews on 100 popular landmarks in Australia. Then, we apply the adapted distant supervision to the reviews and collect 1,000 sentence-triple pairs (we call it the **GEO** test dataset). Table 2 summarizes the statistics of our datasets.

3.3 Joint Learning of Word and Entity Embeddings

Our relation extraction model is based on the encoder-decoder framework which has been widely used in Neural Machine Translation to translate text from one language to another. In our setup, we aim to translate a sentence into triples, and hence the vocabulary of the source input is a set of English words while the vocabulary of the target output is a set of entity and predicate IDs in an existing KG. To compute the embeddings of the source and target vocabularies, we propose a joint learning of word and entity embeddings that is effective to capture the similarity between words and entities for named entity disambiguation (Yamada et al., 2016). Note that our method differs from that of Yamada et al. (2016). We use joint learning by combining skip-gram (Mikolov et al., 2013) to compute the word embeddings and TransE (Bordes et al., 2013) to compute the entity embeddings (including the relationship embeddings), while Yamada et al. (2016) use Wikipedia Link-based Measure (WLM) (Milne and Witten, 2008) that does not consider the relationship embeddings.

Our model learns the entity embeddings by minimizing a margin-based objective function J_E:

$$J_E = \sum_{t_r \in T_r} \sum_{t'_r \in T'_r} \max\left(0, \left[\gamma + f(t_r) - f(t'_r)\right]\right) \quad (1)$$

$$T_r = \{\langle h, r, t\rangle | \langle h, r, t\rangle \in G\} \quad (2)$$

$$T_r{}' = \left\{\langle h', r, t\rangle \mid h' \in E\right\} \cup \left\{\langle h, r, t'\rangle \mid t' \in E\right\} \quad (3)$$

$$f(t_r) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\| \quad (4)$$

Here, $\|\mathbf{x}\|$ is the L1-Norm of vector $\mathbf{x}$, γ is a margin hyperparameter, T_r is the set of valid relationship triples from a KG G, and T'_r is the set of corrupted relationship triples (recall that E is the set of entities in G). The corrupted triples are used as negative samples, which are created by replacing the head or tail entity of a valid triple in T_r with a random entity. We use all triples in Wikidata except those which belong to the testing data to compute the entity embeddings.

To establish the interaction between the entity and word embeddings, we follow the *Anchor Context Model* proposed by Yamada et al. (2016). First, we generate a text corpus by combining the original text and the modified anchor text of Wikipedia. This is done by replacing the entity names in a sentence with the related entity or predicate IDs. For example, the sentence "New York University is a private university in Manhattan" is modified into "Q49210 is a Q902104 in Q11299". Then, we use the skip-gram method to compute the word embeddings from the generated corpus (the entity IDs in the modified anchor text are treated as words in the skip-gram model). Given a sequence of n words $[w_1, w_2, ..., w_n]$, The model learns the word embeddings, by minimizing the following objective function J_W:

$$J_W = \frac{1}{T} \sum_{t=1}^{n} \sum_{-c \le j \le c, j \ne 0} \log P\left(w_{t+j} | w_t\right) \quad (5)$$

$$P\left(w_{t+j}|w_t\right) = \frac{\exp({\mathbf{v}'_{w_{t+j}}}^{\top}\mathbf{v}_{w_t})}{\sum_{i=1}^{W}({\mathbf{v}'_i}^{\top}\mathbf{v}_{w_t})} \quad (6)$$

where c is the size of the context window, w_t denotes the target word, and w_{t+j} is the context word; $\mathbf{v}_w$ and $\mathbf{v}'_w$ are the input and output vector representations of word w, and W is the vocabulary size. The overall objective function of the joint learning of word and entity embeddings is:

$$J = J_E + J_W \quad (7)$$

3.4 N-gram Based Attention Model

Our proposed relation extraction model integrates the extraction and canonicalization tasks for KB enrichment in an end-to-end manner. To build such a model, we employ an encoder-decoder model (Cho et al., 2014) to translate a sentence into a sequence of triples. The encoder encodes a sentence into a vector that is used by the decoder as a context to generate a sequence of triples. Because we treat the input and output as a sequence, We use the LSTM networks (Hochreiter and Schmidhuber, 1997) in the encoder and the decoder.

The encoder-decoder with attention model (Bahdanau et al., 2015) has been used in machine translation. However, in the relation extraction task, the attention model cannot capture the multi-word entity names. In our preliminary investigation, we found that the attention model yields misalignment between the word and the entity.

The above problem is due to the same words in the names of different entities (e.g., the word `University` in different university names such as `New York University`, `Washington University`, etc.). During training, the model pays more attention to the word `University` to differentiate different types of entities of a similar name, e.g., `New York University`, `New York Times Building`, or `New York Life Building`, but not the same types of entities of different names (e.g., `New York University` and `Washington University`). This may cause errors in entity alignment, especially when predicting the ID of an entity that is not in the training data. Even though we add ⟨`Entity-name, Entity-ID`⟩ pairs as training data (see the Training section), the misalignments still take place.

We address the above problem by proposing an n-gram based attention model. This model computes the attention of all possible n-grams of the sentence input. The attention weights are computed over the n-gram combinations of the word embeddings, and hence the context vector for the decoder is computed as follows.

$$\mathbf{c}_t^d = \left[\mathbf{h}^e; \sum_{n=1}^{|N|}\mathbf{W}^n\left(\sum_{i=1}^{|X^n|}\alpha_i^n\mathbf{x}_i^n\right)\right] \quad (8)$$

$$\alpha_i^n = \frac{\exp({\mathbf{h}^e}^{\top}\mathbf{V}^n\mathbf{x}_i^n)}{\sum_{j=1}^{|X^n|}\exp({\mathbf{h}^e}^{\top}\mathbf{V}^n\mathbf{x}_j^n)} \quad (9)$$

Here, $\mathbf{c}_t^d$ is the context vector of the decoder at timestep t, $\mathbf{h}^e$ is the last hidden state of the encoder, the superscript n indicates the n-gram combination, $\mathbf{x}$ is the word embeddings of input sentence, $|X^n|$ is the total number of n-gram token combination, N indicates the maximum value of n used in the n-gram combinations ($N = 3$ in our experiments), $\mathbf{W}$ and $\mathbf{V}$ are learned parameter matrices, and α is the attention weight.

Training

In the training phase, in addition to the sentence-triple pairs collected using distant supervision (see Section 3.2), we also add pairs of ⟨`Entity-name, Entity-ID`⟩ of all entities in the KB to the training data, e.g., ⟨`New York University, Q49210`⟩. This allows the model to learn the mapping between entity names and entity IDs, especially for the unseen entities.

3.5 Triple Generation

The output of the encoder-decoder model is a sequence of the entity and predicate IDs where every three tokens indicate a triple. Therefore, to extract a triple, we simply group every three tokens of the generated output. However, the greedy approach (i.e., picking the entity with the highest probability of the last softmax layer of the decoder) may lead the model to extract incorrect entities due to the similarity between entity embeddings (e.g., the embeddings of `New York City` and `Chicago` may be similar because both are cities in USA). To address this problem, we propose two strategies: re-ranking the predicted entities using a modified beam search and filtering invalid triples using a triple classifier.

The modified beam search re-ranks top-k ($k = 10$ in our experiments) entity IDs that are predicted

Model		WIKI			GEO		
		Precision	Recall	F1	Precision	Recall	F1
Existing Models	MinIE (+AIDA)	0.3672	0.4856	0.4182	0.3574	0.3901	0.3730
	MinIE (+NeuralEL)	0.3511	0.3967	0.3725	0.3644	0.3811	0.3726
	ClausIE (+AIDA)	0.3617	0.4728	0.4099	0.3531	0.3951	0.3729
	ClausIE (+NeuralEL)	0.3445	0.3786	0.3607	0.3563	0.3791	0.3673
	CNN (+AIDA)	0.4035	0.3503	0.3750	0.3715	0.3165	0.3418
	CNN (+NeuralEL)	0.3689	0.3521	0.3603	0.3781	0.3005	0.3349
Encoder-Decoder Models	Single Attention	0.4591	0.3836	0.4180	0.4010	0.3912	0.3960
	Single Attention (+pre-trained)	0.4725	0.4053	0.4363	0.4314	0.4311	0.4312
	Single Attention (+beam)	0.6056	0.5231	0.5613	0.5869	0.4851	0.5312
	Single Attention (+triple classifier)	0.7378	0.5013	0.5970	0.6704	0.5301	0.5921
	Transformer	0.4628	0.3897	0.4231	0.4575	0.4620	0.4597
	Transformer (+pre-trained)	0.4748	0.4091	0.4395	0.4841	0.4831	0.4836
	Transformer (+beam)	0.5829	0.5025	0.5397	0.6181	0.6161	0.6171
	Transformer (+triple classifier)	0.7307	0.4866	0.5842	0.7124	0.5761	0.6370
Proposed	N-gram Attention	0.7014	0.6432	0.6710	0.6029	0.6033	0.6031
	N-gram Attention (+pre-trained)	0.7157	0.6634	0.6886	0.6581	0.6631	0.6606
	N-gram Attention (+beam)	0.7424	**0.6845**	0.7123	0.6816	**0.6861**	0.6838
	N-gram Attention (+triple classifier)	**0.8471**	0.6762	**0.7521**	**0.7705**	0.6771	**0.7208**

Table 3: Experiments result.

by the decoder by computing the edit distance between the entity names (obtained from the KB) and every n-gram token of the input sentence. The intuition is that the entity name should be mentioned in the sentence so that the entity with the highest similarity will be chosen as the output.

Our triple classifier is trained with entity embeddings from the joint learning (see Section 3.3). Triple classification is one of the metrics to evaluate the quality of entity embeddings (Socher et al., 2013). We build a classifier to determine the validity of a triple $\langle h, r, t \rangle$. We train a binary classifier based on the plausibility score $(\mathbf{h} + \mathbf{r} - \mathbf{t})$ (the score to compute the entity embeddings). We create negative samples by corrupting the valid triples (i.e., replacing the head or tail entity by a random entity). The triple classifier is effective to filter invalid triple such as ⟨`New York University, capital of, Manhattan`⟩.

4 Experiments

We evaluate our model on two real datasets including WIKI and GEO test datasets (see Section 3.2). We use precision, recall, and F1 score as the evaluation metrics.

4.1 Hyperparameters

We use grid search to find the best hyperparameters for the networks. We use 512 hidden units for both the encoder and the decoder. We use 64 dimensions of pre-trained word and entity embeddings (see Section 3.3). We use a 0.5 dropout rate for regularization on both the encoder and the decoder. We use Adam (Kingma and Ba, 2015) with a learning rate of 0.0002.

4.2 Models

We compare our proposed model[3] with three existing models including **CNN** (the state-of-the-art supervised approach by Lin et al. (2016)), **MiniE** (the state-of-the-art unsupervised approach by Gashteovski et al. (2017)), and **ClausIE** by Corro and Gemulla (2013). To map the extracted entities by these models, we use two state-of-the-art NED systems including **AIDA** (Hoffart et al., 2011) and **NeuralEL** (Kolitsas et al., 2018). The precision (tested on our test dataset) of AIDA and NeuralEL are 70% and 61% respectively. To map the extracted predicates (relationships) of the unsupervised approaches output, we use the dictionary based paraphrase detection. We use the same dictionary that is used to collect the dataset (i.e., the combination of three paraphrase dictionaries including PATTY (Nakashole et al., 2012), POLY (Grycner and Weikum, 2016), and PPDB (Ganitkevitch et al., 2013)). We replace the extracted predicate with the correct predicate ID if one of the paraphrases of the correct predicate (i.e., the gold standard) appear in the extracted predicate. Otherwise, we replace the extracted predicate with `"NA"` to indicate an unrecognized predicate. We also compare our **N-gram Attention** model with two encoder-decoder based models including the **Single Attention** model (Bahdanau et al., 2015) and **Transformer** model (Vaswani et al., 2017).

[3]The code and the dataset are made available at http://www.ruizhang.info/GKB/gkb.htm

4.3 Results

Table 3 shows that the end-to-end models outperform the existing model. In particular, our proposed n-gram attention model achieves the best results in terms of precision, recall, and F1 score. Our proposed model outperforms the best existing model (MinIE) by 33.39% and 34.78% in terms of F1 score on the WIKI and GEO test dataset respectively. These results are expected since the existing models are affected by the error propagation of the NED. As expected, the combination of the existing models with AIDA achieves higher F1 scores than the combination with NeuralEL as AIDA achieves a higher precision than NeuralEL.

To further show the effect of error propagation, we set up an experiment without the canonicalization task (i.e., the objective is predicting a relationship between known entities). We remove the NED pre-processing step by allowing the CNN model to access the correct entities. Meanwhile, we provide the correct entities to the decoder of our proposed model. In this setup, our proposed model achieves 86.34% and 79.11%, while CNN achieves 81.92% and 75.82% in precision over the WIKI and GEO test datasets, respectively.

Our proposed n-gram attention model outperforms the end-to-end models by 15.51% and 8.38% in terms of F1 score on the WIKI and GEO test datasets, respectively. The Transformer model also only yields similar performance to that of the Single Attention model, which is worse than ours. These results indicate that our model captures multi-word entity name (in both datasets, 82.9% of the entities have multi-word entity name) in the input sentence better than the other models.

Table 3 also shows that the pre-trained embeddings improve the performance of the model in all measures. Moreover, the pre-trained embeddings help the model to converge faster. In our experiments, the models that use the pre-trained embeddings converge in 20 epochs on average, while the models that do not use the pre-trained embeddings converge in $30 - 40$ epochs. Our triple classifier combined with the modified beam search boost the performance of the model. The modified beam search provides a high recall by extracting the correct entities based on the surface form in the input sentence while the triple classifier provides a high precision by filtering the invalid triples.

Discussion

We further perform manual error analysis. We found that the incorrect output of our model is caused by the same entity name of two different entities (e.g., the name of `Michael Jordan` that refers to the American basketball player or the English footballer). The modified beam search cannot disambiguate those entities as it only considers the lexical similarity. We consider using context-based similarity as future work.

5 Conclusions

We proposed an end-to-end relation extraction model for KB enrichment that integrates the extraction and canonicalization tasks. Our model thus reduces the error propagation between relation extraction and NED that existing approaches are prone to. To obtain high-quality training data, we adapt distant supervision and augment it with co-reference resolution and paraphrase detection. We propose an n-gram based attention model that better captures the multi-word entity names in a sentence. Moreover, we propose a modified beam search and a triple classification that helps the model to generate high-quality triples.

Experimental results show that our proposed model outperforms the existing models by 33.39% and 34.78% in terms of F1 score on the WIKI and GEO test dataset respectively. These results confirm that our model reduces the error propagation between NED and relation extraction. Our proposed n-gram attention model outperforms the other encoder-decoder models by 15.51% and 8.38% in terms of F1 score on the two real-world datasets. These results confirm that our model better captures the multi-word entity names in a sentence. In the future, we plan to explore context-based similarity to complement the lexical similarity to improve the overall performance.

Acknowledgments

Bayu Distiawan Trisedya is supported by the Indonesian Endowment Fund for Education (LPDP). This work is done while Bayu Distiawan Trisedya is visiting the Max Planck Institute for Informatics. This work is supported by Australian Research Council (ARC) Discovery Project DP180102050, Google Faculty Research Award, and the National Science Foundation of China (Project No. 61872070 and No. 61402155).

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of Association for Computational Linguistics*, pages 344–354.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. Dbpedia: A nucleus for a web of open data. In *Proceedings of International Semantic Web Conference*, pages 722–735.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of International Joint Conference on Artifical intelligence*, pages 2670–2676.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of International Conference on Neural Information Processing Systems*, pages 2787–2795.

Sergey Brin. 1998. Extracting patterns and relations from the world wide web. In *Proceedings of The World Wide Web and Databases International Workshop*, pages 172–183.

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

Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 1511–1517.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 1724–1734.

Kevin Clark and Christopher D. Manning. 2016. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 2256–2262.

Luciano Del Corro and Rainer Gemulla. 2013. Clausie: clause-based open information extraction. In *Proceedings of International Conference on World Wide Web*, pages 355–366.

Lei Cui, Furu Wei, and Ming Zhou. 2018. Neural open information extraction. In *Proceedings of Association for Computational Linguistics*, pages 407–413.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 1535–1545.

Luis Galárraga, Geremy Heitz, Kevin Murphy, and Fabian M. Suchanek. 2014. Canonicalizing open knowledge bases. In *Proceedings of International Conference on Information and Knowledge Management*, pages 1679–1688.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. Ppdb: The paraphrase database. In *Proceedings of North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764.

Kiril Gashteovski, Rainer Gemulla, and Luciano Del Corro. 2017. Minie: Minimizing facts in open information extraction. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 2620–2630.

Adam Grycner and Gerhard Weikum. 2016. Poly: Mining relational paraphrases from multilingual sentences. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 2183–2192.

Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018. Jointly predicting predicates and arguments in neural semantic role labeling. In *Proceedings of Association for Computational Linguistics*, pages 364–369.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Johannes Hoffart et al. 2011. Robust disambiguation of named entities in text. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 782–792.

Raphael Hoffmann, Congle Zhang, and Daniel S. Weld. 2010. Learning 5000 relational extractors. In *Proceedings of Association for Computational Linguistics*, pages 286–295.

Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2017. Distant supervision for relation extraction with sentence-level attention and entity descriptions. In *Proceedings of AAAI Conference on Artificial Intelligence*, pages 3060–3066.

Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of International Conference on Learning Representations*.

Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. 2018. End-to-end neural entity linking. In *Proceedings of Conference on Computational Natural Language Learning*, pages 519–529.

Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2017. Neural relation extraction with multi-lingual attention. In *Proceedings of Association for Computational Linguistics*, volume 1, pages 34–43.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of Association for Computational Linguistics*, volume 1, pages 2124–2133.

Diego Marcheggiani and Laura Perez-Beltrachini. 2018. Deep graph convolutional encoders for structured data to text generation. *Proceedings of International Conference on Natural Language Generation*, pages 1–9.

Mausam. 2016. Open information extraction systems and downstream applications. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 4074–4077.

Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 523–534.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of International Conference on Neural Information Processing Systems*, pages 3111–3119.

David Milne and Ian H. Witten. 2008. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proceedings of AAAI Workshop on Wikipedia and Artificial Intelligence*, pages 25–30.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of Association for Computational Linguistics and International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *Proceedings of Association for Computational Linguistics*.

Ndapandula Nakashole, Gerhard Weikum, and Fabian M. Suchanek. 2012. Patty: A taxonomy of relational patterns with semantic types. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 1135–1145.

Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Q. Phung. 2018. A novel embedding model for knowledge base completion based on convolutional neural network. In *Proceedings of North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 2, pages 327–333.

Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84.

Christopher De Sa, Alexander Ratner, Christopher Ré, Jaeho Shin, Feiran Wang, Sen Wu, and Ce Zhang. 2017. Incremental knowledge base construction using deepdive. *Very Large Data Bases Journal*, 26(1):81–105.

Wei Shen, Jianyong Wang, and Jiawei Han. 2015. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Trans. Knowl. Data Eng.*, 27(2):443–460.

Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Proceedings of International Conference on Neural Information Processing Systems*, pages 926–934.

Daniil Sorokin and Iryna Gurevych. 2017. Context-aware representations for knowledge base relation extraction. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 1784–1789.

Gabriel Stanovsky, Julian Michael, Ido Dagan, and Luke Zettlemoyer. 2018. Supervised open information extraction. In *Proceedings of North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 885–895.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of International Conference on World Wide Web*, pages 697–706.

Fabian M. Suchanek, Mauro Sozio, and Gerhard Weikum. 2009. SOFIE: a self-organizing framework for information extraction. In *Proceedings of International Conference on World Wide Web*, pages 631–640.

Zequn Sun, Wei Hu, and Chengkai Li. 2017. Cross-lingual entity alignment via joint attribute-preserving embedding. *Proceedings of International Semantic Web Conference*, pages 628–644.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 455–465.

Bayu Distiawan Trisedya, Jianzhong Qi, and Rui Zhang. 2019. Entity alignment between knowledge graphs using attribute embeddings. In *Proceedings of AAAI Conference on Artificial Intelligence*.

Bayu Distiawan Trisedya, Jianzhong Qi, Rui Zhang, and Wei Wang. 2018. Gtr-lstm: A triple encoder for sentence generation from rdf data. In *Proceedings of Association for Computational Linguistics*, pages 1627–1637.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of Neural Information Processing Systems*, pages 5998–6008.

Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Quan Wang, Bin Wang, and Li Guo. 2015. Knowledge base completion using embeddings and rules. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 1859–1865.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint learning of the embedding of words and entities for named entity disambiguation. In *Proceedings of Conference on Computational Natural Language Learning*, pages 250–259.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 1753–1762.

Peng Zhou, Jiaming Xu, Zhenyu Qi, Hongyun Bao, Zhineng Chen, and Bo Xu. 2018. Distant supervision for relation extraction with hierarchical selective attention. *Neural Networks*, 108:240–247.