Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 212–219
Marseille, 11–16 May 2020

Effective Crowdsourcing of Multiple Tasks for Comprehensive Knowledge Extraction

Sangha Nam[1], Minho Lee[1], Donghwan Kim[1], Kijong Han[2], Kuntae Kim[1], Sooji Yoon[1], Eun-kyung Kim[3], Key-Sun Choi[1]
[1]KAIST, [2]Kakao Corp, [3]Daejeon University
{nam.sangha, pathmaker, iedcon, kuntaek, sooji, kschoi}@kaist.ac.kr, matt.han@kakaocorp.com, eunkk@dju.kr

Abstract

Information extraction from unstructured texts plays a vital role in the field of natural language processing. Although there has been extensive research into each information extraction task (i.e., entity linking, coreference resolution, and relation extraction), data are not available for a continuous and coherent evaluation of all information extraction tasks in a comprehensive framework. Given that each task is performed and evaluated with a different dataset, analyzing the effect of the previous task on the next task with a single dataset throughout the information extraction process is impossible. This paper aims to propose a Korean information extraction initiative point and promote research in this field by presenting crowdsourcing data collected for four information extraction tasks from the same corpus and the training and evaluation results for each task of a state-of-the-art model. These machine learning data for Korean information extraction are the first of their kind, and there are plans to continuously increase the data volume. The test results will serve as an initiative result for each Korean information extraction task and are expected to serve as a comparison target for various studies on Korean information extraction using the data collected in this study.

Keywords: Information Extraction, Crowdsourcing, Natural Language Processing, Knowledge Base

1. Introduction

The recent victory of IBM Watson (Ferrucci, 2012) over human competitors in a quiz show provides a new impetus to almost all fields of artificial intelligence, such as natural language processing (NLP), question answering, knowledge representation, extraction, and reasoning. Research on knowledge extraction is currently underway to extract structured knowledge from unstructured text in the form of Resource Description Framework (RDF) triples. Various large-scale knowledge bases such as DBpedia (Auer et al., 2007), YAGO (Suchanek et al., 2007), and Wikidata (Vrandečić and Krötzsch, 2014) are widely used in many NLP tasks. Knowledge extraction has two main phases for enriching the knowledge base: entity linking and relation extraction. In the entity linking step, entities that can be linked to a knowledge base are identified from natural language sentences. Then the relation extraction step follows, where relations between entities are classified and knowledge is extracted from natural language sentences in a structured data format. Entity linking and relation extraction are indispensable for knowledge extraction from natural language text. They are often accompanied by coreference resolution to maximize the efficiency and recall of knowledge extraction.

With recent advances made in deep learning technology, high-performance deep learning architectures such as convolutional neural networks (CNN) and long short-term memory (LSTM) have emerged and been used in all phases of knowledge extraction. A large amount of training data needs to be fed into these neural network architectures. Training data have been generated by experts, but this method is extremely cost-intensive and time-consuming. To overcome the drawbacks of expert-driven training data generation, crowdsourcing-based data collection has recently been increasingly applied. With the crowdsourcing method, a vast amount of data is gathered from an undefined large group of the general public and refined to be used as input data for training deep learning neural network models. In particular, Snow et al. (Snow et al., 2008) showed that, for many NLP tasks, crowdsourced data is as good as or better than that annotated by experts.

Many previous studies have used shared data or publicized their crowdsourcing data (Liu et al., 2016; Chamberlain et al., 2016; Demartini et al., 2012), but they have the drawback of lacking a corpus that enables a sequential training and evaluation process consisting of entity linking, coreference resolution, and relation extraction from the same document. Although those data were constructed in the same language (i.e., English), they performed tests and evaluations with data collected from different document in each step, such as entity linking from a twitter data and relation extraction from Wikipedia or the new york times corpus. This makes it difficult to analyze the overall knowledge extraction performance according to the error propagation from one step to the next.

In this paper, we describe a process of generating crowdsourcing data to execute the four tasks(entity mention detection, entity linking, co-reference resolution, and relation extraction) of knowledge extraction using the same corpus and presents the results of training with a state-of-the-art model based on the collected data. Our data can test the entity mention detection and linking, coreference resolution, and relation with the same document, then it can be conducted a full performance test from the knowledge extraction perspective. In addition, the effect of the error generated in the previous step on subsequent steps can be tested also. For example, in our experience, we found that errors in entity linking have an adverse effect on learning relation extraction model. Therefore, when creating crowdsourcing data, we decided that using a common source text rather

than using different source texts for each task would be helpful for the overall analysis of information extraction. These crowdsourcing data for Korean knowledge extraction are the first of their kind, and there are plans to continuously increase the volume of the data. Important considerations for crowdsourcing data collection are 1) whether proper quality control has been conducted; 2) more data have been collected at a lower cost compared to collection by an expert group; and 3) whether the collected data can be applied to a machine learning model. We sought to ensure quality management and cost reduction in every task by adopting the gated instruction (GI) protocol presented by Liu et al. (Liu et al., 2016) and setting up a machine—human collaboration process. We also created various tools to facilitate the tasks of crowd workers. Details are described in Section 3. Sections 4 and 5 present the process of applying Korean text data to a state-of-the-art model for each task using the collected data. The experimental results served as the initiative result for each Korean knowledge extraction task and are expected to serve as the comparison target for various studies on Korean knowledge extraction based on the proposed data. Our proposed data set and evaluation results will be a Korean knowledge extraction initiative point and promote research in this field.

2. Background and Related Work

2.1. Entity Linking

Entity linking is a task of linking a specific entity *e* to a mention *m* from a knowledge base, where the entity set *E* and a natural language text with the mention set *M* represent entities. For example, consider the sentence "Steve Jobs is Apple's founder," where ["Steve Jobs", "Apple"] is given as the mention set *M* of this text. Then, the entity linking task is to correctly link *m* "Apple" to "Apple_(company)" instead of "Apple_(fruit)." This task is conducted in two sub-steps. First, the entity linking model detects entity candidates that may match the mentions to be linked in the knowledge base. Then, the entity to be linked is searched from the set of entity candidates through a comparison of the candidates' linking scores. Recent models are related to word embedding, context word embedding, and entity embedding implemented with the output of the entity description (Gupta et al., 2017); modeling a mention set assuming latent relations among entities (Le and Titov, 2018); and a mixed model of jointly learning mention detection and entity linking (Kolitsas et al., 2018).

2.2. Coreference Resolution

Because a previously mentioned word (antecedent) is usually referred to in various other forms in a natural language (e.g., a pronoun, demonstrative determiner, or abbreviation), understanding the text is necessary to verify whether such expressions are referring to the co-referring entity. This process of grouping all expressions referring to the co-referring entity is termed coreference resolution. It plays an important role in knowledge extraction because entity linking alone is insufficient for catching all information. For example, consider knowledge extraction from the text "Gordon Moore majored in electrical engineering. He was born in the United States." Without coreference resolution, only the triple "`Field`(Gordon_Moore, Electrical_Engineering)" can be extracted because entity linking cannot extract the triple "`birthPlace`(Gordon_Moore, United_States)" since it is not known which entity "he" is referring to.

Coreference resolution has recently been extensively explored in studies related to deep learning (Clark and Manning, 2016; Wiseman et al., 2016). In particular, Lee et al. (Lee et al., 2018) achieved an F1 score as high as 73% with English coreference resolution based on a deep learning model. The basic architecture of this model is composed of two parts. First, it searches the representations of all possible spans (candidate mentions) occurring in a document using Bi-LSTM and computes the mention score, which indicates the likelihood of a candidate mention being an actual mention. Then, it computes the antecedent score which indicates the anaphoric relation between two spans, and completes the coreference resolution task by combining the mention and antecedent scores. They addressed the problem of a coreference occurring with a word in between (regardless of singular or plural) by a higher-order span representation using an attention mechanism, and they employed a coarse-to-fine method to reduce the computation load.

2.3. Relation Extraction

Relation extraction refers to the task of extracting the relation between two entities in a sentence. For example, a relation extraction system will extract "`Founder`(Facebook, Mark Zuckerberg)" from the sentence "Mark Zuckerberg is the founder of Facebook." Traditional approaches rely heavily on human intervention through handcrafted rules and hand-tagged training data of pre-specified relations. Distant supervision (DS) learning has been used for relation extraction in almost all studies since it was introduced by Mintz et al. (Mintz et al., 2009). DS approach automatically collects training data by pairing the knowledge and associated sentence. Many studies have used the DS approach to expand target relations and reduce the cost of constructing handcrafted training data. However, a statistical analysis of the DS data from Wikipedia-DBpedia collected in this study was found to contain 49% noise. For example, "`Founder`(Steve Jobs, Apple)" from the sentence "Steve Jobs argued with Wozniak, the co-founder of Apple." Despite intensive research efforts to remove noise automatically (Han and Sun, 2016; Hoffmann et al., 2011; Jiang et al., 2016; Surdeanu et al., 2012; Zhou et al., 2016), this problem has yet to be solved. In particular, some groups (Angeli et al., 2014; Liu et al., 2016; Pershina et al., 2014; Zhang et al., 2012) have attempted to boost the performance of relation classifiers through crowdsourcing-based training data collection for relation extraction and using them along with training data for DS learning.

Recent research on relation extraction has focused on the generative adversarial network (GAN) or reinforcement learning with more complicated architectures. Wu et al. (Wu et al., 2017) explored GAN-based relation extraction by using perturbed embedding and adding noise to the pretrained word embedding value. Recent approaches to re-

lation extraction based on reinforcement learning have also attracted much attention. Zeng et al. (Zeng et al., 2018) proposed a model for predicting the bag representation from a bag of sentences, which is a set of sentences containing two same target entities, where the relation extractor plays the role of an agent. To develop models for directly solving the noisy data problem of training data for DS learning, Feng et al. (Feng et al., 2018) and Qin et al. (Qin et al., 2018) proposed architectures consisting of a sentence selector to remove noisy-labeled sentences from the training text and a relation extractor. Through reinforcement learning, the sentence selector acts as an agent and is trained to maximize the reward output from the relation extractor. This improves the performance of the relation extractor because noisy sentences are filtered in training data.

2.4. Crowdsourcing

Research into crowdsourcing-based machine learning has gained traction (Inel et al., 2014; Snow et al., 2008). Machine learning models for NLP tasks using data generated by the general public with common sense–level knowledge have been shown to be equivalent or superior to models involving expert-generated data. This has encouraged many researchers to generate massive training data for various NLP tasks with the crowdsourcing approach and apply them to machine learning models.

Bontcheva et al. (Bontcheva et al., 2017) and Demartini et al. (Demartini et al., 2012) conducted crowdsourcing based research on entity linking. They collected web pages to extract entity mentions and presented the entity candidate set for each entity mention to the crowd workers with the request to select one from among the entity candidates. Bontcheva et al. (Bontcheva et al., 2017) proposed a method of providing the abstract (definition) of each entity candidate. In a study on crowdsourced coreferences, Chamberlain et al. (Chamberlain et al., 2016) conducted the annotation in two steps: presentation of a coreferent mention within a document (e.g., pronoun, demonstrative determiner, or abbreviation) and antecedent annotation by a crowd worker; and quality control by two crowd workers to indicate whether the correct antecedent was annotated.

Liu et al. (Liu et al., 2016) improved the relation extraction model's classification performance by using data collected with the GI protocol, which is a crowdsourcing scheme that they designed. The GI protocol consists of three phases: in the tutorial, workers are trained with annotation practice and immediate feedback on the same UI, in weed-out, workers are tested with simple questions and disqualified if they fail to solve them; in annotation, workers are given batches of gold question sets created by an expert, and further participation is granted only to those whose annotations show high agreement ($\geq$80%) with the answers provided by the expert. Data generation by a worker trained according to the GI protocol without duplicate data allocations can significantly boost the classification performance of a relation extraction model compared with data generation by multiple workers through joint work and majority approval.

We set up a four-phase crowdsourcing scheme for knowledge extraction and applied the GI protocol to all phases.

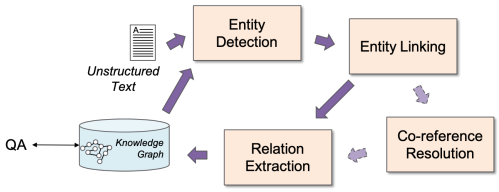

Figure 1: Overview of the knowledge extraction and crowdsourcing data collection process

The next section describes the schemes in detail.

3. Design

We target Wikipedia-style text. As source data, we extracted paragraphs available for DS data collection and previous paragraphs. For example, if DS data were collected from paragraph number 5 of the Wikipedia document number 3, paragraphs number 1 to 5 in the document number 3 were selected as source data for crowdsourcing to generate data for knowledge extraction at the paragraph and document levels, not simply at the sentence level.

Fig. 1 shows a conceptual diagram illustrating the data collection workflow implemented in this study. The output of each phase was used as the input of the next phase, which enabled a more coherent and complete knowledge extraction in comparison with separate data collection for each phase. In Phase I (entity mention detection annotation), the worker read the given text and selected candidate mentions for the entity. In Phase II (entity linking annotation), candidate mentions from Phase I that could be linked to the knowledge base were annotated. In Phase III (coreference annotation), the pronoun and demonstrative determiners in the given text and the antecedent of the new entity identified in entity mention detection were searched. In Phase IV (relation extraction annotation), clues to the relationship between two entities in the given text were examined. The GI protocol was employed for all tasks, and each worker started annotating after being trained in a tutorial and passing the sample test. The reliability (annotation accuracy) of the participating workers was assessed by having them answer the trap questions prepared by an expert every 7–10 hits. Quality management was ensured by eliminating the previous 7–10 hits of a worker with a cutoff score. A worker who repeatedly failed to pass the test with trap questions was disqualified. In each phase, automatic tools were introduced to provide hints that may reduce the difficulty level. These tools were described in detail in each phase.

3.1. Entity Mention Detection

Entity mention detection is the task of searching for an entity mention in the given text. Fig. 2 shows the annotation interface layout for entity mention detection. The source text is displayed paragraph-wise in the upper left quarter. In the case of Wikipedia, the worker was provided with text tagged with the related Wikilink information (green-shaded words). This enabled the worker to intuitively understand which mention should be tagged additionally (blue-shaded words) while reading the text by checking the sample entities in real time. After selecting the entity mention, the

Figure 2: Annotation interface for entity mention detection; All examples of annotation interface are in Korean. However, All annotation interfaces can be applied directly to all other languages.

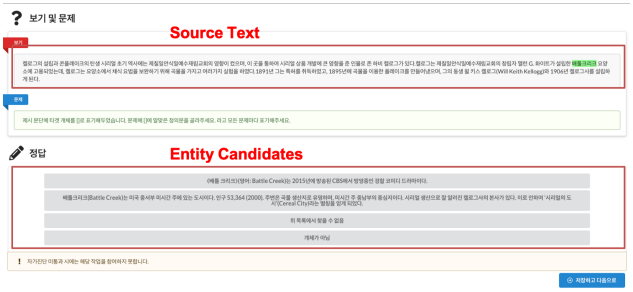

Figure 3: Annotation interface for entity linking

worker selected the entity category from among 16 categories (person, study field, theory, artifact, organization, location, civilization, event, year, time, quantity, job, animal, plant, material, and term) typically used for named entity recognition tasks. We paid 0.25$ per paragraph for this work, and each paragraph was assigned to two workers to minimize the odds of an entity mention being missed. Entity mention detection used trap questions to continuously assess the quality of work, while simultaneously assigning the same data to two different workers. The reason for assigning the work to two people is that even a skilled worker may not find all the entities perfectly. Second, even the same entity may tagged with different grade and types of entities depending on context and worker. Thus, the final data is created by merging the results of two people's work.

3.2. Entity Linking

Entity linking is the task of linking the entity mentions in the given text to their respective entities in a knowledge base. Fig. 3 shows the annotation interface layout for entity linking. The text and green-shaped entity mentions were displayed in the uppermost part of the screen, and the worker selected one of the abstracts (or descriptions) of individual entity candidates provided underneath as potential options. The abstracts were provided to help the worker better understand the context and increase the chance of the right answer being selected. Each set of options had "Not in candidate" and "Not an entity" in addition to the abstracts of the individual entity candidates; the worker was to choose the former if he/she could not find any adequate abstract and the latter for an erroneously tagged entity. Entity candidates were provided as a result of an automatic search among the entities in a knowledge base. Besides the main reason for providing "Not in candidate" and "Not an entity" (i.e., to collect accurate data), these choices, particularly the former, were provided to establish a dataset for the entity discovery task to register a new entity in the knowledge base. We paid 0.15$ per paragraph for the work in this phase, and each paragraph was assigned to one worker.

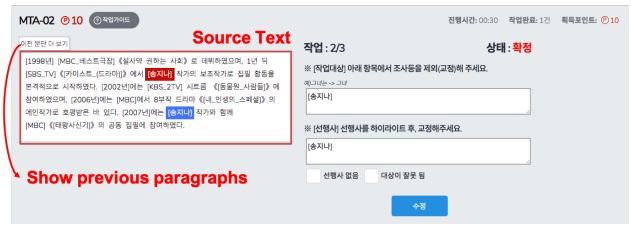

Figure 4: Annotation interface for coreference resolution

3.3. Coreference Resolution

General-purpose coreference resolution is known to be a difficult NLP task. The highest known performance in English is 73% (Lee et al., 2018) and the performance is much lower in Korean at 58% (Park et al., 2018). Given this relatively low performance level of coreference resolution systems, the error propagation to the next phase of relation extraction is likely to be high. Furthermore, the current representative data generation/annotation guidelines for general-purpose coreference resolution are complicated and have many rules to follow (Park et al., 2014; Pradhan et al., 2012). Even setting the boundary of a mention is governed by complicated rules, as well as judging whether two mentions are co-referring. This entails difficulties in data annotation by the (nonprofessional) general public. To overcome this problem, we scaled down the coreference resolution by limiting the scope of the target mentions to a named entity, pronoun, and definite noun phrase necessary for knowledge extraction. The scope was narrowed down to these three elements because other elements did not need to be considered for knowledge extraction. A named entity is the core element for knowledge extraction. A pronoun and definite noun phrase are typically used to refer to an antecedent and allow intuitive and simple judgement about the scope of mentions and their coreference. For this reason, for example, previous studies on Korean coreference resolution have also limited the scope of mentions to these elements (Choi et al., 2007; Park and Lee, 2017).

Fig. 4 shows the annotation interface layout for coreference resolution. First, mentions were extracted as the basis for rules. Named entities were received from the entity linking phase, and pronouns and definitive noun phrases were extracted automatically with 99% recall owning to simple detection rules. On the annotation interface, automatically extracted mentions were highlighted as shown in the upper left quarter. With this interface, the worker annotated data by clicking a mention that could be an antecedent or the option "No antecedent" or "Entity error". If coreference resolution was impossible because of a lack of an antecedent in the given paragraph, the worker could click the button (upper left quarter) "Show previous paragraphs" to open all

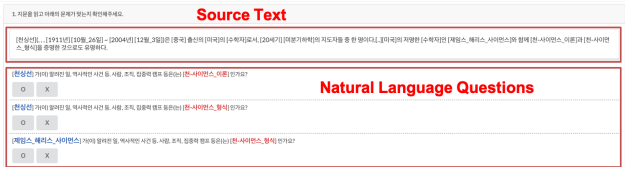

Figure 5: Annotation interface for relation extraction

Table 1: Entity mention detection and linking corpus

Dataset	Ours-Crowdsourcing	Ours-Goldset	AIDA/CoNLL
Document	2,574	434	1,393
Avg. words in dataset	200	39	216
Total number of mentions	152,301	4,186	34,956
Not-in-candidate	10,566	137	7,136
Not-an-entity	1,836	35	-
Empty candidate	17,096	-	-

previous paragraphs and tag a matching antecedent. This is an essential feature for coreference resolution at the document level. We paid 0.15$ per paragraph for the work in this phase, and each paragraph was assigned to one worker.

3.4. Relation Extraction

Relation extraction is a task of identifying relations between entity pairs in the given text. Fig. 5 shows the annotation interface layout for relation extraction. Unlike the previous phases, where co-recurring entities were searched and grouped together, relation extraction was a process by which the relations between entities tagged in previous tasks were captured. We collected a relation extraction dataset in accordance with the method assuming DS (Mintz et al., 2009) and expanded from the sentence level to the paragraph level to capture entity pairs occurring in different sentences. Each relation (property) used for DS collection had short English labels as defined in the ontology schema. Because the property could be interpreted differently by different workers, we provided it in the format of a natural language yes/no question. The meaning of each property was taken from the Wikidata description information. For example, we converted `birthPlace` to "birth location of a person, animal or fictional character." We paid 0.4$ per hit (15 questions on average) for the work in this phase, and each set was assigned to one worker.

4. Resources

We built crowdsourcing data from Korean Wikipedia and KBox (Nam et al., 2018). As far as we know, Amazon Mechanical Turk and Crowdflower are widely used in English. However, there is not a Korean related work there, and it is not easy to gather Korean worker. Therefore, we use Crowdworks[1] platform, which is the representative of Korean crowdsourcing companies. Ontological knowledge extraction requires a reference knowledge base. This knowledge base is indispensable for entity linking and relation extraction and plays a pivotal role in extracting correct information that meets the schema of the knowledge base. For the reference knowledge base, we selected KBox (Nam et al., 2018), which is an expansion of the Korean DBpedia.

[1]https://www.crowdworks.kr

Table 2: Coreference resolution corpus

Dataset	Ours-Crowdsourcing	Ours-Goldset	CoNLL-2012
Document	2,660	207	3,395
Sentence	55,406	2,493	90,191
Mention of chain (A)	109,484	3,839	187,384
Reference chain (B)	32,172	1,127	40,355
Ratio of A/B	3.403	3.406	4.643

Table 3: Relation extraction corpus

Dataset	Ours-Crowdsourcing	Ours-Goldset
Source	Wiki-KBox	Wiki-KBox, News-KBox
No. of relation	114 (a)	76 (subset of a)
No. of true labeled data	40,091	3,170
No. of false labeled data	39,330	0

Table 1 presents the basic statistics of the entity detection and linking dataset that we collected and those of AIDA/CoNLL dataset (Hoffart et al., 2011), which is widely used in the English-speaking world. We collected a total of 2,574 document-level data using the crowdsourcing method presented in Section 3, which was nearly double the size of AIDA/CoNLL. To test the crowdsourcing model, we built 434 expert-reviewed document-level gold data with fewer words per document compared with the training data. As mentioned previously, mentions for which no adequate definitions could be found were classified as not-in-candidate, and erroneously tagged entities were classified as not-an-entity. An empty candidate was defined as a mention for which no entity candidates were found automatically.

Table 2 presents the basic statistics of the coreference resolution dataset that we collected and those of CoNLL-2012 dataset (Pradhan et al., 2012), which is widely used in English. As mentioned previously, our coreference resolution dataset was based on source data with entities that were tagged by workers in the entity mention detection and linking tasks to indicate whether there were antecedents in the same document; pronoun and demonstrative determiner candidates were detected with a pronoun extractor. We had 1,480 document-level data, which was slightly less than half the size of CoNLL-2012. We built a gold set reviewed by four experts for model testing when the crowdsourcing task was completed. The mention of a chain (A) refers to the number of mentions with antecedents, and a reference chain (B) to the actual number of entities referred to the mentions by grouping. The ratio between the mention of a chain and reference chains per document of CoNLL was 4.6, and that of our model was 3.3; this means that one entity was mentioned approximately 3.3 times on average within a document.

Table 3 presents the basic statistics of the relation extraction dataset collected in this study. The number of extracted relations in our data was 114, which is more than twice those of TAC-KBP (41) and NYT10 (51), which are commonly used in English. This dataset allowed the relations of entity pairs in different sentences to be identified with paragraph-level DS. Source data were collected from Korean Wikipedia and KBox, and the average noise was calculated to be 49.5%; in particular, deathPlace (97%) and birthPlace (96%) had so much noise that a model trained

Table 4: Evaluation results for entity linking

Model	Precision	Recall	F1 score
KO	0.93	0.91	0.92
EN	0.88	0.98	0.93

with DS data alone would not be able to discern the relation between them. We use more Wikipedia corpus than common corpus in entity linking and co-reference resolution. Because when we collected the crowdsourcing data only with the common corpus of entity linking and co-reference resolution, the dataset was small and the relation extraction model could not be trained. We use all first paragraphs in Korean Wikipedia pages. Four experts of the Telecommunications Technology Association [2] generated 3,190 relation extraction gold data to test the relation extraction model with Korean Wikipedia and New Corpus.

5. Evaluation

5.1. Entity Linking

For the entity linking model, we used the one proposed by Le and Titov (Le and Titov, 2018), which we adapted to the Korean language. This model uses the candidate mentions that can be extracted from a document mention M to assess the potential relation between the context and candidate. Let there by K number of relations and candidate set C_i be generated to match mention $m_i \in M$. Then, the score of each candidate is calculated by the local scoring model with the attention mechanism for the context word of m_i to match $c_{ij} \in C_i$. In the pairwise scoring model, the entity pair score is calculated by applying K number of relation matrices per entity candidate to the (m_i, m_j) pair. The obtained candidate-context score and candidate-candidate score are then used to derive the entity set with the highest score in the document. We used the methods presented by Le and Titov (Le and Titov, 2018) (300-dimension GloVe) and Gupta et al. (Gupta et al., 2017) to create word embedding and entity embedding, respectively, to adapt this model to the Korean language.

To test the model performance, we used the crowdsourcing data discussed in Section 4 for training. Table 4 outlines the assessment results with the gold set. The model showed an excellent performance (F1 score: 95%), which indirectly demonstrates the high quality of the data used in this study.

5.2. Coreference Resolution

For the coreference resolution model, we used the one proposed by Lee et al. (Lee et al., 2018). We modified the model as follows to adapt it to the Korean language. 1) The token-level representations of the input vector were broken down into morpheme-level representations. 2) The three word embedding models (Word2Vec, ELMo, and Character Embedding) were retrained with the Korean Wikipedia corpus. 3) Entities known from entity linking take additional mention scores. 4) The NER type was added to each mention as a feature.

Table 5 outlines the performances of the coreference resolution model trained with the crowdsourcing data. The

[2]https://www.tta.or.kr

Table 5: Evaluation results for coreference resolution

	MUC			B^3			CEAF			
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Avg. F1
KO	0.88	0.65	0.75	0.86	0.60	0.71	0.81	0.64	0.72	0.72
EN	0.83	0.61	0.71	0.81	0.55	0.66	0.77	0.55	0.65	0.67

Table 6: Evaluation results for relation extraction models

Architecture	Embedding	Precision	Recall	F1 score
PCNN-DS	FastText	0.62	0.55	0.58
PCNN-Crowd	FastText	0.77	0.58	0.66
GAN-DS	Skip-gram	0.67	0.63	0.65
GAN-Crowd	Skip-gram	0.75	0.59	0.66
GAN-DS	FastText	0.58	0.59	0.58
GAN-Crowd	FastText	0.74	0.63	0.68
GAN-DS	ELMo	0.70	0.64	0.67
GAN-Crowd	ELMo	0.76	0.70	0.73
RL-DS	FastText	0.80	0.67	0.73
RL-Crowd	FastText	0.82	0.73	0.77
BERT-DS	BERT-base, Multilingual Cased	0.88	0.76	0.82
BERT-Crowd	BERT-base, Multilingual Cased	**0.89**	**0.78**	**0.83**

average F1 score of the Korean version was slightly higher than that of the English version (0.72 vs. 0.67). As noted above, however, our model has a downscaled mention detection system.

5.3. Relation Extraction

We tried various methods to apply the typically used English relation extraction model to Korean. First, we compared the performances of the word embedding models when trained with Skip-gram, FastText, and ELMo (Peters et al., 2018). Then, we applied the collected data to four most recent relation extraction architectures: PCNN (Zeng et al., 2015), GAN (Wu et al., 2017), RL (Feng et al., 2018), and BERT (Soares et al., 2019). Of the 114 relations collected by crowdsourcing, we used 49 relations for training and testing. We selected those with a noise ratio not exceeding 50% to check whether the model performance improved when crowdsourcing data were added to the DS noisy data. Thus, we used 228,096 and 20,603 DS labeled data and crowdsourcing data, respectively, for these 49 relations. Table 6 presents the test results.

The postfix "DS" attached to an architecture model name indicates that the model was trained only with DS data, and the postfix "Crowd" indicates that the model was trained with crowdsourcing data in addition to DS data. The test results can be summarized as follows:

1. The performance of all models improved by an F1 score of up to 10% when crowdsourcing data were added to DS data.

2. The classification performance of a model increased as the embedding method became increasingly context-oriented (from Skip-gram to FastText and ELMo).

3. In the inter-model performance comparison, the F1 score increased with the architecture complexity in the order of PCNN, GAN, RL, and BERT, which required increasingly long training runs.

Detailed descriptions of the model architecture and hyperparameters are omitted from this paper due to limited space. There are no significant changes compared with the original paper except for the embedding size.

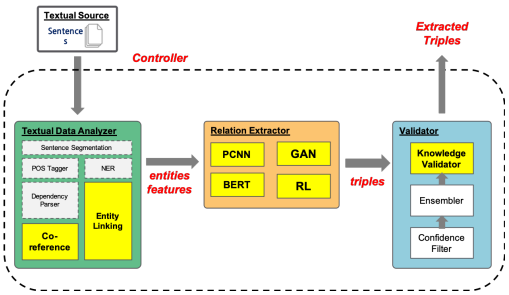

Figure 6: Overview of knowledge extraction model

5.4. Knowledge Extraction

We developed a knowledge extraction system by combining the previous models that learned with our crowdsourcing data as shown in Fig. 6. We evaluated the accuracy of the randomly selected 150 extracted triples manually. The overall accuracy is 0.49. As can be seen from the results, the knowledge extraction performance is lower than the performance of each model for each task. This happens when the error in the previous model propagates to the next model. For example, because the entity linking system links the word 'director' as a film director instead of a sports team manager, the system finally extract an error triple "`occupation`(Guus_Hiddink, Film_director)". Another case that causes error is the problem of extracting a relation even though there is no relationship between the two entities (not a relation problem). For example, an error triple "`occupation`(Soccer_player, Manager)" is extracted from a sentence "Jin soon-jin is a retired Korean soccer player and a current manager of Chungwoon high school." As such, our current data can be used to perform error analysis step-by-step on the results of the knowledge extraction system output.

6. Conclusion

In this study, we collected crowdsourcing data for Korean knowledge extraction tasks and presented the performance test results of the state-of-the-art model with the collected data. The presented data allowed all tasks involved in knowledge extraction (i.e., entity linking, co-reference resolution, and relation extraction) to be evaluated with the same source data to facilitate reading comprehension. We plan to develop a framework for performing comprehensive knowledge extraction analysis in the future. There are also plans to collect data for a corpus of zero-anaphora resolutions characteristic of the Korean language. The data and source code can be found in the GitHub repository [3] , and additional data and evaluation code will be continuously updated at the Github and the Hackathon[4].

7. Acknowledgements

This work was supported by Institute of Information Communications Technology Planning Evaluation(IITP) grant funded by the Korea government(MSIT) (2013-0-00109, WiseKB: Big data based self-evolving knowledge base and reasoning platform) This work was supported by the Industrial Strategic technology development program (10072064, Development of Novel Artificial Intelligence Technologies To Assist Imaging Diagnosis of Pulmonary, Hepatic, and Cardiac Diseases and Their Integration into Commercial Clinical PACS Platforms) funded by the Ministry of Trade Industry and Energy (MI, Korea)

[3]https://github.com/machinereading/crowdsourcing
[4]http://www.okbqa.org

8. Bibliographical References

Angeli, G., Tibshirani, J., Wu, J., and Manning, C. D. (2014). Combining distant and partial supervision for relation extraction. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1556–1567.

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.

Bontcheva, K., Derczynski, L., and Roberts, I. (2017). Crowdsourcing named entity recognition and entity linking corpora. In *Handbook of Linguistic Annotation*, pages 875–892. Springer.

Chamberlain, J., Poesio, M., and Kruschwitz, U. (2016). Phrase detectives corpus 1.0 crowdsourced anaphoric coreference. In *LREC*.

Choi, M., Lee, C., Wang, J., and Jang, M.-G. (2007). Reference resolution for ontology population. In *Proceedings of the 19th Annual Conference on Human and Cognitive Language Technology*.

Clark, K. and Manning, C. D. (2016). Improving coreference resolution by learning entity-level distributed representations. *arXiv preprint arXiv:1606.01323*.

Demartini, G., Difallah, D. E., and Cudré-Mauroux, P. (2012). Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st international conference on World Wide Web*, pages 469–478. ACM.

Feng, J., Huang, M., Zhao, L., Yang, Y., and Zhu, X. (2018). Reinforcement learning for relation classification from noisy data. In *Proceedings of AAAI*.

Ferrucci, D. A. (2012). Introduction to "this is watson". *IBM Journal of Research and Development*, 56(3.4):1–1.

Gupta, N., Singh, S., and Roth, D. (2017). Entity linking via joint encoding of types, descriptions, and context. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2681–2690.

Han, X. and Sun, L. (2016). Global distant supervision for relation extraction. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Hoffart, J., Yosef, M. A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., and Weikum, G. (2011). Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 782–792. Association for Computational Linguistics.

Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L., and Weld, D. S. (2011). Knowledge-based weak supervision for information extraction of overlapping relations. In

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pages 541–550. Association for Computational Linguistics.

Inel, O., Khamkham, K., Cristea, T., Dumitrache, A., Rutjes, A., van der Ploeg, J., Romaszko, L., Aroyo, L., and Sips, R.-J. (2014). Crowdtruth: Machine-human computation framework for harnessing disagreement in gathering annotated data. In *International Semantic Web Conference*, pages 486–504. Springer.

Jiang, X., Wang, Q., Li, P., and Wang, B. (2016). Relation extraction with multi-instance multi-label convolutional neural networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1471–1480.

Kolitsas, N., Ganea, O.-E., and Hofmann, T. (2018). End-to-end neural entity linking. In *CoNLL*.

Le, P. and Titov, I. (2018). Improving entity linking by modeling latent relations between mentions. *arXiv preprint arXiv:1804.10637*.

Lee, K., He, L., and Zettlemoyer, L. (2018). Higher-order coreference resolution with coarse-to-fine inference. *arXiv preprint arXiv:1804.05392*.

Liu, A., Soderland, S., Bragg, J., Lin, C. H., Ling, X., and Weld, D. S. (2016). Effective crowd annotation for relation extraction. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 897–906.

Mintz, M., Bills, S., Snow, R., and Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Nam, S., Kim, E.-k., Kim, J., Jung, Y., Han, K., and Choi, K.-S. (2018). A korean knowledge extraction system for enriching a kbox. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 20–24.

Park, C. and Lee, C. (2017). Coreference resolution for korean pronouns using pointer networks. *Journal of KIISE*, 44(5):496–502.

Park, C.-E., Choi, K.-H., and Lee, C. (2014). Korean coreference resolution using the multi-pass sieve. *Journal of KIISE*, 41(11):992–1005.

Park, C., Lee, C., Ryu, J., and Kim, H. (2018). Contextualized embedding and character embedding-based pointer network for korean coreference resolution. In *Proceedings of the 30th Annual Conference on Human and Cognitive Language Technology*.

Pershina, M., Min, B., Xu, W., and Grishman, R. (2014). Infusion of labeled data into distant supervision for relation extraction. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 732–738.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., and Zhang, Y. (2012). Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint Conference on EMNLP and CoNLL-Shared Task*, pages 1–40. Association for Computational Linguistics.

Qin, P., Xu, W., and Wang, W. Y. (2018). Robust distant supervision relation extraction via deep reinforcement learning. *arXiv preprint arXiv:1805.09927*.

Snow, R., O'Connor, B., Jurafsky, D., and Ng, A. Y. (2008). Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 254–263. Association for Computational Linguistics.

Soares, L. B., FitzGerald, N., Ling, J., and Kwiatkowski, T. (2019). Matching the blanks: Distributional similarity for relation learning. *arXiv preprint arXiv:1906.03158*.

Suchanek, F. M., Kasneci, G., and Weikum, G. (2007). Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM.

Surdeanu, M., Tibshirani, J., Nallapati, R., and Manning, C. D. (2012). Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 455–465. Association for Computational Linguistics.

Vrandečić, D. and Krötzsch, M. (2014). Wikidata: a free collaborative knowledge base.

Wiseman, S., Rush, A. M., and Shieber, S. M. (2016). Learning global features for coreference resolution. *arXiv preprint arXiv:1604.03035*.

Wu, Y., Bamman, D., and Russell, S. (2017). Adversarial training for relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1778–1783.

Zeng, D., Liu, K., Chen, Y., and Zhao, J. (2015). Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762.

Zeng, X., He, S., Liu, K., and Zhao, J. (2018). Large scaled relation extraction with reinforcement learning. *Relation*, 2:3.

Zhang, C., Niu, F., Ré, C., and Shavlik, J. (2012). Big data versus the crowd: Looking for relationships in all the right places. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 825–834. Association for Computational Linguistics.

Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., and Xu, B. (2016). Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 207–212.