Automatic Ontology Population Using Deep Learning for Triple Extraction

Ming-Hsiang Su, Chung-Hsien Wu and Po-Chen Shih
Department of Computer Science and Information Engineering,
National Cheng Kung University, Taiwan
E-mail: {huntfox.su, chunghsienwu}@gmail, bobshih@hotmail.com

***Abstract*— Ontology is a kind of representation used to represent knowledge in a form that computers can derive the content meaning. The purpose of this work is to automatically populate an ontology using deep neural networks for updating an ontology with new facts from an input knowledge resource. In this study for automatic ontology population, a bi-LSTM-based term extraction model based on character embedding is proposed to extract the terms from a sentence. The extracted terms are regarded as the concepts of the ontology. Then, a multi-layer perception network is employed to decide the predicates between the pairs of the extracted concepts. The two concepts (one serves as subject and the other as object) along with the predicate form a triple. The number of occurrences of the dependency relations between the concepts and the predicates are estimated. The predicates with low occurrence frequency are filtered out to obtain precise triples for ontology population. For evaluation of the proposed method, we collected 46,646 sentences from Ontonotes 5.0 for training and testing the bi-LSTM-based term extraction model. We also collected 404,951 triples from ConceptNet 5 for training and testing the multilayer perceptron-based triple extraction model. From the experimental results, the proposed method could extract the triples from the documents, achieving 74.59% accuracy for ontology population.**

I. INTRODUCTION

An ontology is a kind of representation of knowledge, which is machine-interpretable and searchable. Nowadays, there are many ontology languages describing the ontology, and one of the widely used ontology languages is the Resource Description Framework (RDF), which is defined and promoted by World Wide Web Consortium [1]. In this ontology language, N-Triples is one of the formats popularly used for representing an RDF graph [2]. In an ontology, the knowledge is composed of triples, each of which is represented as a sequence of (**subject**, **predicate**, **object**) terms and the **predicate** describes the relation between the two concepts, **subject** and **object**. The concepts could be words or phrases. There are many researchable issues related to ontology [3]-[4], such as ontology enrichment [5], ontology population [6], and inconsistency resolution [7]. This work focused triple extraction for ontology population.

There are some reasons why researchers study the ontology [8]. First, the process of studying the ontology is to analyze the domain knowledge, and researchers expect that the ontology contains the domain knowledge which can be further used by the application systems. After having the specific domain ontology, systems can extract and gather information according to the ontology structure. In addition, systems can also apply the ontology in other aspects. For example, if there are many websites containing medical information and this information is defined by the same ontology, systems can gather the data from these websites and apply the collected data to other aspects. In a situation that predecessors have researched some domain knowledge and had a completed ontology, it can directly use the ontology they designed when the other domain knowledge crosses the predecessors' research. Furthermore, to construct a huge ontology, we can combine or populate the other existing ontologies.

In chatbot systems, it has been proven that ontologies are helpful and effective [9]-[10] for dialog state tracking [11]-[13]. In Spoken Language Understanding (SLU), Jang et al. [9] showed that the accuracy could be improved after using the ontology. In Dialog State Tracking (DST), Mehta et al. [10] constructed a decision tree to determine the user intent with the help of the ontology. Using the ontology to construct the decision tree is more convincing and accurate, and the performance of tracking the whole dialogue is also better. Therefore, ontology is useful for a chatbot system [14]-[15]. However, constructing an ontology is very difficult and time-consuming. Therefore, this work adopted the predicates defined by ConceptNet which focuses on people's common cognition and stores the common sense in the form of ontology.

In the progress of ontology population, we need to firstly extract the specific words as the concepts of a triple. This task is like the Named Entity Recognition (NER) task. Therefore, we utilize the technology of NER to obtain the knowledge. Nowadays, most studies have applied recurrent neural networks (RNN) as the sequence-labeling model with the character-level embedding or other word features in the NER task [16]-[17]. Lample et al. [17] used a Conditional Random Field (CRF) layer to decide the current tag after the RNN, which was different from the previous research. Due to the feature of CRF, the current tag considered both the current output vector and the previous tag. As a result, it improved the recognition accuracy. On the other hand, they also changed the character-level embedding which concatenated the pre-trained word vector and the output vectors of a character-level bidirectional long-short term memory (Bi-LSTM). In this work, we extracted the key term based on a Bi-LSTM.

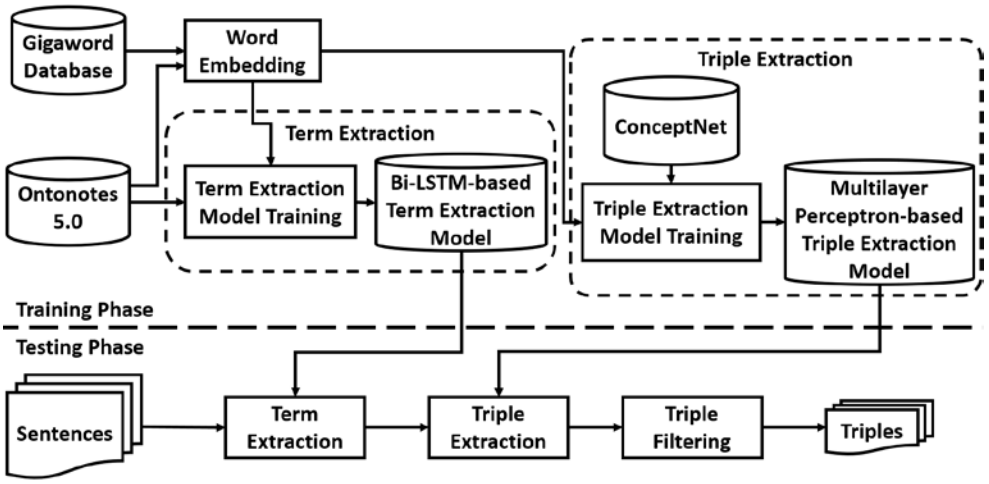

Fig. 1 *Schematic diagram of DNN-based ontology population system.*

For constructing an ontology with new facts, populating the ontology is one of the most important steps in which the goal is to add new triples into the current ontology for updating. Fernandez and Ponnusamy [18] designed a specialized user interface for the aimed ontology for Non-Governmental Organization (NGO). According to the interface, users could add the leader's name, the founding time, the organization purpose, etc. Makki [19] used Part-of-Speech tag to create rules and applied the rules to the documents to populate the ontology of risk management. These rules were called Hearst Patterns [20]. In this work, we proposed a deep neural network (DNN)-based method for automatic ontology population.

Fig. 1 shows the proposed system framework of the DNN-based ontology population system. The proposed method of the DNN-based ontology population system has two main parts: the Bi-LSTM-based term extraction model and the multilayer perceptron (MLP)-based triple extraction model. First, we train the Bi-LSTM-based term extraction model by using the Ontonotes 5.0 database. We retrieve the extracted terms which may become the part of the candidate triples. Then, these paired terms are sent to the MLP-based triple extraction model to determine if the predicates could be selected. Finally, the triple extraction model is to check whether the pair of subject and object has a predicate that can describe their relationship. The extracted (**subject**, **predicate**, **object**) terms are then used for ontology population.

II. Database Collection

There are two databases used in this work, the named entity database in Ontonotes 5.0 [21] and ConceptNet 5 [22]. Ontonotes 5.0 includes various genres of text, including news, conversational telephone speech, weblogs, broadcast, talk shows, etc. There are three languages in Ontonotes 5.0, English, Chinese and Arabic. The labeled data in Ontonotes 5.0 includes named entity, coreference and structural data such as parsing tree and predicate argument structure. There are 18 named entity types, such as Person, Organization and Location, in Ontonotes5.0. We only use the Chinese named entity part in Ontonotes 5.0 and there are 46,645 sentences in the Chinese corpus. Word segmentation is performed by Jieba word segmentation tool [23] without modifying the tags.

ConceptNet is designed to help computers understand the meanings of words that people use. The knowledge in ConceptNet includes words and phrases, and it describes not only the definitions in dictionaries but also the common sense, such as {台灣(Taiwan) IsA 地區(Region)} describing the Taiwanese attributes, and {台灣(Taiwan) SymbolOf 珍珠奶茶(Bubble Milk Tea)} describing the related facts of Taiwan. Totally, there are about 28 million triples in ConceptNet. As for predicates, there are 29 positive predicates and 4 negative predicates. We removed the triples in which each subject or object consists of more than two words; for example {一個小小的島(A small island)} having multiple words will be removed. Totally, we use 23 predicates consisting of 163,727 Chinese triples, and one of 23 predicates is negative predicate.

III. Term Extraction Model

In this study, we train the Bi-LSTM as the term extraction model. The Bi-LSTM is used to train the sequence dependencies by updating the weight matrices of input gate, forget gate and output gate. The Bi-LSTM produces the output vector with the current input vector and the previous output vector. The CRF layer uses the output vector to learn the weight of each feature function.

If there are a potential sequence $\mathbf{I} = (\boldsymbol{i_1}, \ldots, \boldsymbol{i_T})$ with a length of T and a corresponding observation sequence $\mathbf{O} = (\boldsymbol{o_1}, \ldots \boldsymbol{o_T})$, the CRF is to obtain the maximum probability of (1). In the equation, there are two features for each time step, transition feature and state feature. Then, (1) can be expanded into (3).

$$\mathrm{P(I|O)} = \frac{1}{Z(O)} e^{\Sigma_t^{\mathrm{T}} \Sigma_k^{\mathrm{M}} \lambda_k * f_k(i_{t-1}, i_t, o_t)} \tag{1}$$

$$\mathrm{Z(O)} = \sum_{I} e^{\Sigma_t^{\mathrm{T}} \Sigma_k^{\mathrm{M}} \lambda_k * f_k(i_{t-1}, i_t, o_t)} \tag{2}$$

$$P(I|O) = \frac{1}{Z(O)} e^{(\sum_t^T \sum_m^M \alpha_m * tran_m(i_{t-1}, i_t, o_t) + \sum_t^T \sum_n^N \alpha_n * stat_n(i_t, o_t))} \quad (3)$$

The subscript t means the t^{th} element. The value of t is from 1 to T, the length of a sequence. The subscript k means the k^{th} feature function, and each feature function has a corresponding weight, λ_k, which is updated while training. Each feature function can be split into a transition function and a state function. At time *t*, there are *M* transition functions and *N* state functions. $tran_m$ indicates the m^{th} transition function, and $stat_n$ indicates the n^{th} state function. When satisfying the condition of the feature function, the function will return 1, otherwise return 0. (4) is an example of the state function.

$$stat_0(i_0, o_0) = \begin{cases} 1, current\ named\ entity\ is\ Person \\ 0, others \end{cases} \quad (4)$$

The training procedure of the CRF has two steps. The first step is to produce feature functions from the training data and initialize the weight of each function. The second step is to use the optimization methods, like maximum likelihood estimation and gradient descent, to update the feature function weights until the weights converge.

For CRF, the output sequence of the Bi-LSTM is the observation sequence of CRF, and we can calculate the gradient with the predictions of CRF and the term sequence. Using back propagation algorithm, we update the weights in Bi-LSTM and CRF. Finally, the trained model is used to detect the key terms in a sentence.

For term extraction, each sentence is segmented into several words by Jieba Chinese word segmentation tool [23], and the term extraction model decides which words form a term. The term extraction model we used is a Bi-LSTM [17], which considers the word-level and character-level embeddings, with a CRF layer to enhance the reliability of the results. The word-level embedding uses the traditional embedding method, word2vec [24], and the character-level embedding uses an another Bi-LSTM to retrieve the forward and backward vectors as the character-level vector. The word-level vector is trained by the Skip-Gram model of word2vec with the Gigaword database [25]. According to [24] which proposed the word2vec model, though the term frequency of the rarely used words is smaller than that of the commonly used words, the vectors trained by the Skip-Gram model have good performance in experiments. Because the term frequencies are relatively small in the large corpus, we use the Skip-Gram to train the word-level vectors. The second part of word vector is the character-level vector which is modified as the features of Chinese language. As shown in Fig. 2, we segment the sentences into words, and these words are the inputs of the Bi-LSTM, which serves as the character-level embedding model, to encode the characters into two vectors [17]. For example, $F_{台北}$ means the forward vector, and $B_{台北}$ means the backward vector. Finally, the word vector of 台北 (Taipei) is the concatenation of $E_{台北}$, $F_{台北}$ and $B_{台北}$.

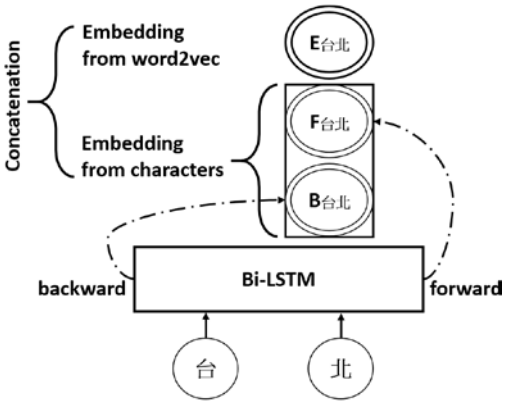

Fig. 2 *Schematic diagram of Character-Level embedding of [17].*

IV. Triple Extraction Model

After term extraction, we need to decide the predicates between every two terms. As there are 23 predicates used in this study, 23 MLPs, each deciding if there was a predicate between two terms, are used. The outputs of each perceptron model are the confidence scores.

As shown on the left of Fig. 3, a single perceptron follows (5) to compute the result Y_j. $\boldsymbol{w}_i$ is the weight matrix of the perceptron. φ is the activation function, such as the rectified linear unit (ReLU) activation function. As shown on the right of Fig. 3, a single layer perceptron gathers many perceptrons. Each single layer perceptron can produce different responses to the signal. A single layer perceptron only deals with the linear-separable problems. As for the linear-inseparable problems, we need to stack many single layer perceptrons to form an MLP. Because the problem of predicate classification was linearly inseparable, we use an MLP as the classifier

$$Y_j = \varphi\left(\sum_i \boldsymbol{w}_i x_i + \mathrm{b}\right) \quad (5)$$

Fig. 4 shows the framework of triple extraction model. Since we think there are different features between subjects and objects, we separate the embedding of the subjects and objects. The first and second layers are used to encode and compress the word vectors. The third and fourth layers are used to interpret the concatenated vector. Finally, the model outputs the confidence scores for this candidate triple. We use the existing triples in the ConceptNet to train the 23 MLP-based triple extraction models.

V. Experiments and Results

For evaluation, we collected 46,646 sentences from Ontonotes 5.0 for evaluating the Bi-LSTM-based term extraction model and collected 404,951 triples from ConceptNet 5 for evaluating MLP-based triple extraction model. We used five-fold cross validation method to conduct the following experiments. The results were the mean of the experiments and the data in the parenthesis was the standard deviations. The models included the Bi-LSTM-based term extraction model which was trained by using Ontonotes

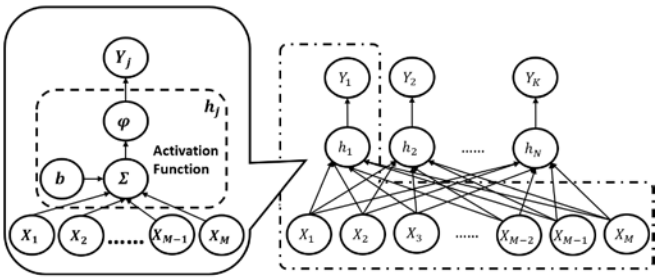

Fig. 3 Framework of a single layer perceptron with many perceptrons.

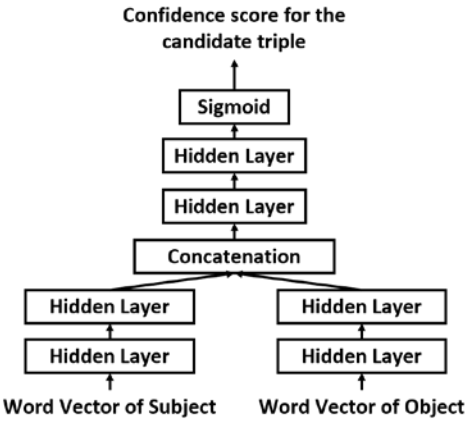

Fig. 4 Framework of triple extraction model.

database and the MLP-based triple extraction model which was trained by using triples in ConceptNet.

A. *Experimental results of term extraction*

Since we considered a sentence as the minimum unit for ontology population, the first step was to extract the key terms in a sentence. There were 37,316 sentences for training and 9,329 sentences for testing. The number of vocabularies was 693,160 words (20,870 are distinct). The experiment used 25 dimensions for character-level embedding, and 300 dimensions for the word vectors which were pre-trained by Gigaword and Ontonotes database. The experiments used the Cangjie and the Pinyin Chinese input system as the character-level embedding method. Cangjie [26], based on the graphological aspect of the characters, is a system by which Chinese characters may be entered into a computer using a standard keyboard, while Pinyin is the official Romanization system for Standard Chinese. In addition, we considered the fusion at decision level and feature level to improve the results. We evaluated the performance by F1 score. In this experiment, we evaluated the system performance on character embedding and different input method embeddings, such as Cangjie and Pinyin. The experimental results showed that the character embedding method achieved the best performance, as shown in Table I. Although the performance of Cangjie embedding was worse than character embedding, Cangjie embedding provided a new aspect for character-level embedding in Chinese and the results were comparable to character embedding.

Table I Comparisons on different embeddings for term extraction

Method	F1 score
Character embedding	**85.20 (0.34)**
Cangjie embedding	85.02 (0.24)
Pinyin embedding	84.97 (0.28)
Decision level fusion (Cangjie + Pinyin)	84.17 (0.32)
Feature level fusion (Cangjie + Pinyin)	84.96 (0.36)

B. *Experimental results of triple extraction model*

The second step to populate the ontology was to determine the predicates between the terms in pairs. The experimental results of the proposed model and the Neural Tensor Network (NTN) model [27] are shown in Table II. According to the results, the proposed models performed better than the NTN-based triple extraction models. In 23 predicate models, the NTN-based triple extraction models outperformed the proposed method for only 5 predicates, which were Antonym, HasA, SimilarTo, SymbolOf and EtymologicallyRelatedTo. As the results, we believed that if there were enough training data, our proposed models could work better than the NTN-based triple extraction models.

Table II The experimental results of triple extraction models.

Method	Positive Triples	Accuracy
Proposed method	346,571	**84.62%**
NTN method	346,571	81.94%

Although we had the triple extraction model for each predicate, it was still not reliable when they were applied to real application. To avoid our models from producing many incorrect triples, we proposed two methods to remove the unreliable triples. First, we analyzed the term distribution from the surface text in ConceptNet with our term extraction model. Only the triples in which the subjects and objects were in the top 3 of the term occurrence frequency were fed to the triple extraction model. Second, we used E-HowNet, the people name list and the company name list in Wikipedia to define the rules. We removed incorrect triples based on the defined rules. Finally, Table III shows the number of triples and accuracy that our system populated with/without term frequency filtering and rule filtering after reading 1,268 documents. According to the experiments, using the term frequency and rules to remove incorrect triples was useful, and improved the accuracy obviously.

Table III The number of populated triples and accuracy.

Method	Triple count	Correct triples	Accuracy
No filter method	1,798	239	13.29%
Method 1	513	175	34.11%
Method 2	185	138	**74.59%**

Method 1: with term distribution filtering.
Method 2: with term distribution filtering and rule filtering.

VI. Conclusions and Future Work

In this work, we proposed an approach to ontology population. We tried to use two new character-level embedding methods, the Cangjie and the Pinyin, in term extraction model. Although the results were only comparable to the character embedding model, we presented new aspects for character

embedding in Chinese. For triple extraction, the proposed models performed better than the NTN models. In addition, compared to the original methods to populate the ontology with manual definition rules by observing the features of languages or documents, the proposed method is relatively easy to populate correct triples.

In the future, we are planning to modify the character-level embedding method based on the Cangjie and the Pinyin, to improve the term extraction model performance. We are also planning to improve the predicate recognition model by considering the context of a sentence. In the meanwhile, there is no database consisting of the sentences with tags defined in the ontology. Currently, only the database like ConceptNet can be used, and this is also an aspect where we are going to work on.

REFERENCES

[1] J. Z. Pan, "Resource Description Framework," in *Handbook on Ontologies*, S. Staab, and R. Studer, Eds., Heidelberg: Springer Berlin Heidelberg, 2009, pp. 71-90.

[2] D. Beckett, G. Carothers, and A. Seaborne, "RDF 1.1 N-Triples, A line-based syntax for an RDF graph," *W3C Recommendation*, February 2014. Available: https://www.w3.org/TR/2014/REC-n-triples-20140225/.

[3] P. Buitelaar, P. Cimiano, and B. Magnini, "Ontology learning from text: An overview," in *Ontology Learning from Text: Methods, Evaluation and Applications*, vol. 123, P. Buitelaar, P. Cimmiano, and B. Magnini, Eds. IOS Press, Amsterdam, 2005, pp. 3-12.

[4] G. Petasis, V. Karkaletsis, G. Paliouras, A. Krithara, and E. Zavitsanos, "Ontology population and enrichment: State of the art," in *Knowledge-driven multimedia information extraction and ontology evolution*, G. Paliouras, C. D. Spyropoulos, and G. Tsatsaronis, Eds., Heidelberg: Springer-Verlag Berlin, 2011, pp. 134-166.

[5] S. Castano, I. S. E. Peraldi, A. Ferrara, V. Karkaletsis, A. Kaya, R. Möller, S. Montanelli, G. Petasis, and M. Wessel, "Multimedia interpretation for dynamic ontology evolution," *Journal of Logic and Computation*, vol. 19, no. 5, pp. 859-897, September 2008.

[6] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates, "Web-scale information extraction in knowitall:(preliminary results)," *Proc. of the 13th international conference on World Wide Web*, pp. 100-110, May 2004.

[7] P. Haase, F. Van Harmelen, Z. Huang, H. Stuckenschmidt, and Y. Sure, "A framework for handling inconsistency in changing ontologies," *Proc. of international semantic web conference*, pp. 353-367, November 2005.

[8] N. F. Noy and D. L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology," 2001. Available: http://www-ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness-abstract.html.

[9] Y. Jang, J. Ham, B.-J. Lee, Y. Chang, and K.-E. Kim, "Neural dialog state tracker for large ontologies by attention mechanism," *Proc. of Spoken Language Technology Workshop* (*SLT*), pp. 531-537, December 2016.

[10] N. Mehta, R. Gupta, A. Raux, D. Ramachandran, and S. Krawczyk, "Probabilistic ontology trees for belief tracking in dialog systems," *Proc. of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 37-46, September 2010.

[11] M. H. Su, C. H. Wu, K. Y. Huang, and C. K. Chen, "Attention-Based Dialog State Tracking for Conversational Interview Coaching," *Prof. of 2018 IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), pp. 6144-6148, April 2018.

[12] M. H. Su, K. Y. Huang, T. H. Yang, K. J. Lai, and C. H. Wu, "Dialog State Tracking and action selection using deep learning mechanism for interview coaching," *Proc. of 2016 International Conference on Asian Language Processing* (*IALP*), pp. 6-9, November 2016.

[13] M. H. Su, C. H. Wu, K. Y. Huang, T. H. Yang, and T. C. Huang, "Dialog state tracking for interview coaching using two-level LSTM," *Proc. of 2016 10th International Symposium on Chinese Spoken Language Processing* (*ISCSLP*), pp. 1-5, October 2016.

[14] C. H. Wu, M. H. Su, and W. B. Liang, "Miscommunication handling in spoken dialog systems based on error-aware dialog state detection," *EURASIP Journal on Audio, Speech, and Music Processing*, no. 2017: 9, pp. 1-17, May 2017.

[15] M. H. Su, C. H. Wu, K. Y. Huang, and W. H. Lin, "Response Selection and Automatic Message-Response Expansion in Retrieval-Based QA Systems using Semantic Dependency Pair Model," *ACM Transactions on Asian and Low-Resource Language Information Processing* (*TALLIP*), vol. 18, no. 1, pp. 3:1-3:24, January 2019.

[16] J. P. Chiu and E. Nichols, "Named entity recognition with bidirectional LSTM-CNNs," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 357-370, December 2015.

[17] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," *Proc. of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 260-270, June 2016.

[18] F. M. H. Fernandez and R. Ponnusamy, "Automated populates and updates personalized ontology with analysis result," *Proc. of IEEE International Conference on Advanced Communications, Control and Computing Technologies*, pp. 580-585, May 2014.

[19] J. Makki, "Ontoprima: A Prototype for Automating Ontology Population," *International Journal of Web & Semantic Technology*, vol. 8, no. 4, October 2017.

[20] M. A. Hearst, "Automatic acquisition of hyponyms from large text corpora," *Proc. of the 14th Conference on Computational Linguistics-Volume 2*, pp. 539-545, August 1992.

[21] S. Pradhan, A. Moschitti, N. Xue, O. Uryupina, and Y. Zhang, "CoNLL-2012 shared task: Modeling Multilingual Unrestricted Coreference in OntoNotes," *Proc. of Conference on Computational Natural Language Learning '12 Joint Conference on EMNLP and CoNLL - Shared Task*, pp. 1-40, July 2012.

[22] R. Speer and C. Havasi, "Representing General Relational Knowledge in ConceptNet 5," *Proc. of the 8th international conference on Language Resources and Evaluation (LREC)*, pp. 3679-3686, May 2012.

[23] J. Sun, "Jieba Chinese word segmentation tool," 2012. Available: https://github.com/fxsjy/jieba.

[24] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Proc. of the 27th Advances in Neural Information Processing Systems*, pp. 3111-3119, December 2013.

[25] J. F. Hong, and C. R. Huang, "Using Chinese Gigaword Corpus and Chinese Word Sketch in Linguistic Research," *Proc. of the 20th Pacific Asia Conference on Language, Information and Computation (PACLIC)*, pp. 183-190, November 2006.

[26] R. Kang, H. Zhang, W. Hao, K. Cheng, and G. Zhang, "Learning Chinese Word Embeddings with Words and Subcharacter N-Grams," *IEEE Access*, vol. 7, pp. 42987-42992, March 2019.

[27] R. Socher, D. Chen, C. D. Manning, and A. Ng, "Reasoning with neural tensor networks for knowledge base completion," *Proc. of the 27th Advances in neural information processing systems*, pp. 926-934, December 2013.