Automatic Knowledge Graph Construction: A Report on the 2019 ICDM/ICBK Contest

Xindong Wu*†, Jia Wu‡, Xiaoyi Fu*, Jiachen Li*, Peng Zhou†§, and Xu Jiang*
*Mininglamp Academy of Sciences, Mininglamp Technology, Beijing 10084, China
†Key Laboratory of Knowledge Engineering with Big Data (Hefei Univ. of Technology), Ministry of Education, Hefei, China
‡Department of Computing, Macquarie University, Sydney, NSW 2109, Australia
§School of Computer Science and Technology, Anhui University, Hefei 230601, China
Emails: wuxindong@mininglamp.com, jia.wu@mq.edu.au, fuxiaoyi@mininglamp.com,
lijiachen@mininglamp.com, doodzhou@hotmail.com, jiangxu@mininglamp.com

***Abstract*—Automatic knowledge graph construction seeks to build a knowledge graph from unstructured text in a specific domain or cross multiple domains, without human intervention. IEEE ICDM 2019 and ICBK 2019 invited teams from both degree-granting institutions and industrial labs to compete in the 2019 Knowledge Graph Contest by automatically constructing knowledge graphs in at least two different domains. This article reports the outcomes of the Contest. The participants were expected to build a model to extract knowledge represented as triplets from text data and develop a web application to visualize the triplets. Awards were given to five teams. Their models and key techniques used to construct knowledge graphs are summarized.**

I. Introduction

Knowledge graphs are prevalent in applications like web search [1], recommendation [2], and question answering [3] and, currently, most high-quality knowledge graph projects are built by volunteers through crowd-sourcing, *e.g., Wikidata* [4]. A system that could automatically construct a knowledge graph would vastly accelerate this laborious process and, as a result, knowledge could be structured and managed in many more business scenarios. One real-world example is building a domain-specific knowledge graph from the news articles of a vertical industry website, which could then be used to empower intelligent downstream applications.

How to construct knowledge graphs from text has been a challenging problem for years [5]. The predominant approach [6], [7] is to develop a pipeline of NLP tasks, such as named entity recognition followed by co-reference resolution, entity linking, and then relation extraction. This approach scores well against standard classification evaluation metrics, such as precision, recall, and F1, but few efforts have paid attention to how human knowledge is structured. It is difficult to accurately reflect, in advance, the relations that will appear in someone's mind when they read a piece of text.

A. Data Graph vs. Knowledge Graph

As of 18 September 2019, Wikipedia (https://en.wikipedia.org/wiki/Knowledge_Graph) refers knowledge graph as *a knowledge base used by Google and its services to enhance its search engine's results with information gathered from a variety of sources*. There are nowadays a large number of companies who have been building their knowledge graphs to support various tasks and functions. However, 99% of existing "knowledge graphs" are actually data graphs without knowledge. Taking "Bob and I were high-school classmates, and I will invite him for a dinner to celebrate our 25th year class reunion in 2020" as an example, if a graph cannot identify who is "him" and does not provide any support on when they graduated from high school, the graph is only a data graph.

We define a knowledge graph as a semantic graph for describing concepts and their relations in a physical world with three essential components:

1) Concepts. A concept can be an entity (such as a person), an attribute (such as age), or a fact (such as "a red car with 4 doors"), and is represented as a node.
2) Relations. A relation is a connection between two nodes with a semantic label, such as "is-a", "has-a" or action (*e.g.,* "becomes").
3) Background knowledge about concepts and relations. A concept can have different names, such as Professor X. Wu and Dr. Xindong Wu, and possibly multiple attributes such as height and occupation, and a relation can have different appearances such as "had", "has" and "have". The background knowledge in the form of a dictionary or an ontology can semantically link different names, attributes and appearances.

When there is no background knowledge about nodes or relations, a graph with nodes and connections is a data graph. The two basic constructs of a knowledge graph are "entity-relation-entity" triplets and "entity-attribute" pairs. In both these constructs, entities are connected by their relations to form a graph-structured knowledge base.

Given that knowledge graphs were originally designed to enhance search engines, it is unsurprising that the state-of-the-art performances tend to be associated with web search. Through a knowledge graph, the Web can be transformed from a collection of hyperlinks into a set of concept links, which means users can retrieve information about subjects based on

2374-8486/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDM.2019.00204

true semantics instead of rudimentary character matching.

B. Challenges in Knowledge Graph Construction

Let us start with two text examples.

1. Bob hit the nail into the wall with a hammer.

2. John had a new fast 4-wheel car, and the car became a slow one 2 years later.

The challenges in knowledge graph construction are 3-fold:

1) information loss,
2) information redundancy, and
3) information overlapping.

Information loss occurs when the output graph is incomplete. Information redundancy means extra concepts and relations that do not exist in the input text but in the background knowledge. For example in the above sentence 1, a complete and accurate depiction includes the following entities: *Bob*, *nail*, *wall* and *hammer*, and the following relations: *(Bob, hit, nail)*, *(nail, into, wall)*, and *(Bob, with, hammer)*.

In sentence 2, we have entity 'the car' which was changing from a fast one to a slow one in two years. In this regard, the information overlapping challenge refers to whether a knowledge graph can encode the changing of an attribute.

II. The 2019 ICDM/ICBK Contest

The purpose of the Contest was to generate a knowledge graph of the thought patterns of a person when reading a piece of text. To this end, the "person" was assumed to be a human reader in the subject matter. Hence, the competition was judged by human experts. Given that different experts inevitably pay attention to different components of the text, the veracity with which a knowledge graph reflects a person's thought patterns is somewhat subjective. To ensure as much objectivity as possible, each contest submission was evaluated by two experts[1]. The final shortlist was decided by the organizing committee after examining all the package materials submitted by each team.

Each participating team tested their approach on the same dataset, which was assembled by the contest organizers. The dataset consists of 300 published news articles evenly distributed across four different industries: automotive engineering, cosmetics, public security, and catering services. Each article is between 150 to 250 words in length and contains 8-20 entities. Also, the articles relating to each industry were reviewed by domain experts as a group to ensure that the diversity and depth of knowledge is representative of the real world and that the language standard of each article is appropriate, neither too formal nor too poorly written.

120 articles from the 300 article dataset were manually labeled in advance by multiple experts from the above four industries. Entity words that are synonymous were grouped and labeled by the experts to form a synonym dictionary at the article level, followed by manual labeling of semantic relations between pairs of entity words. At the online evaluation stage, each entity word in a submission was first replaced with a synset label according to this synonym dictionary. Then, the labels of entity words mentioned in each article were compared with those labeled by the experts to achieve the purpose of synonym tolerance. The standard of fault tolerance of each article was determined by the industry experts responsible for labeling.

Each team was invited to build a model that takes an article as input and output a graph. The following conditions were stipulated: the nodes must be entity words or phrases from the article; the edges must be relation words or phrases between entities; and the nodes must be denoted by words or phrases from the original text, and synonyms of the same word should be merged.

Similar competitions for building a graph representation of knowledge from open text have been held in NLP forums in the past, but in these contests, a predefined schema of entities and/or relations was given in advance for subsequent extraction with an information extraction model. The novelty of this Contest is that no schema of any kind was given in advance for either entities or relations.

III. Key Techniques in Constructing Knowledge Graphs

A typical knowledge-graph construction process consists of three main components: information extraction, knowledge fusion, and knowledge processing. This Contest only involves information extraction and knowledge fusion.

The objective of information extraction is to identify and separate entities in a data source, along with the attributes of these entities and their relations to other entities. Therefore, information extraction is an apt name for this process since no actual "knowledge" is output directly in this step. The two main technologies involved in information extraction are entity recognition and relation extraction. Additionally, co-reference resolution will be involved in knowledge fusion.

A. Entity Recognition

Entity extraction, also known as named entity recognition (NER), refers to the process of identifying accurately named entities from data, especially text data [8]. The three main classes are: entity class (such as person name, place name, and institution name), time class (such as date and time), and number class (*e.g.,* currency and percentage) [9]. Note that these classes can be expanded to suit specific application domains.

NER technology has undergone a transition from rule-based methods to statistical approaches. The following paradigms have been developed for NER activities.

1) Rule-based approaches: In early efforts on NER, especially at the Message Understanding Conferences, the basic idea behind most mainstream NER methods was to manually construct a limited set of rules and then search for strings in the text that matched these rules. In later studies, researchers looked to automatically discover and generate the rules with machines.

[1]Note that the online evaluation tool described in the contest information at the conference website that was available to all participants was only to help them track their model iterations during model development.

2) Machine learning-based approaches: NER studies based on machine learning can be roughly divided into three themes: the selection of models and methods, the improvement of models and methods, and the selection of features.

3) Deep learning-based approaches: In recent years, deep learning technology has become a new research hotspot in machine learning and, like in many other areas, and it has been successfully tasked to solve some NER problems. In particular, word vector representation has provided a powerful driving force for the typical serialized labeling problems of NER. The simplest and, arguably, most effective approach to using word vectors to represent features was proposed in [10]. Distributed word representations for Twitter micro-posts with NER were proposed in [11]. More recently, [12] proposed a neural semi-Markov structured SVM model to control the trade-off between precision and recall by assigning weights to different types of errors in the loss-augmented inference process during training.

The winning teams in this Contest have used a range of approaches for entity recognition.

- **Team UWA** used the NLP tool SpaCy [13] to classify words with corresponding part-of-speech (POS) tags and chunked noun and verb phrases extracted according to predefined rules. A noun chunk is defined as the words describing the noun, and a verb chunk is defined as the verb and its surrounding adpositions or particles [14]. In the visualization step, the noun chunks are assigned to the same category as the most similar entity recognized by SpaCy, and the categories of nodes are color-coded.
- **Team Tmail** extracted named entities with the Stanford OpenIE toolkit [15], OpenIE 5.0 [16], and SpaCy [13]. They correct these entities to their original words from the data that were modified by the OpenIE toolkit.
- **Team BUPT-IBL** used their own model SC-LSTM [17] in addition to Stanford CoreNLP [15] and SpaCy [13]. To remove redundant entities as a result of using two extraction models, they designed a string matching rule.
- **Team MIDAS-IIITD** used NLTK [18] and SpaCy [13] for preprocessing. Then the NLP toolkit flair [19] was used to split sentences into chunked phrases, some of which were selected to construct the output triplets.
- **Team Lab1105** used SpaCy [13]. As a supplement, they trained a BiLSTM+CRF model on the CoNLL 2003 NER dataset [20], which contains four types entities: persons (PER), organizations (ORG), locations (LOC), and miscellaneous names (MISC).

B. Relation Extraction

Once the entities (nodes) in a graph have been obtained through information extraction, the next step is relation extraction for edge construction.

Similar to entity extraction methods, most early efforts on relation extraction were based on rules. However, progress with rule-based methods was limited. Performance has improved since researchers began to incorporate supervised learning into what is now considered state-of-the-art. Meanwhile, supervised learning methods require an abundance of manually labeled samples, which generally means enormous labor costs. Semi-supervised, unsupervised, and self-supervised methods have been developed to reduce the need for labels. Although these methods have resulted in some progress toward model versatility, natural languages are very complex, and so the relation extraction problem is far from being solved. An overview of the main learning paradigms [21] for relation extraction follows.

1) Supervised learning methods: Supervised learning methods embody the idea of classification on human-annotated data. Once trained, these methods can recognize entities through a matching process and extract specific relations. Supervised learning for relation extraction can be divided into two main categories: feature vector-based methods and kernel-based methods.

2) Semi-supervised methods: Most semi-supervised methods have two additional steps compared to the above supervised approaches. The first is to pre-design a few relation types. The second is to incorporate appropriate entity pairs as seeds into training sets. These methods alleviate dependence on huge amounts of labels.

3) Domain-independent methods: Domain-independent methods relax the need for domain specifications, which means these methods are easy to extend and can even be applied to multiple domains. Some researchers have incorporated external knowledge bases, such as Wikipedia, to complement their methods [22]. [23] proposed a framework for open information extraction and an extraction model named TextRunner, while [24], [25] improved TextRunner's performance. These methods assume that each pair of entities has a known relation and uses context information to construct feature representations for the entities.

4) Distant-supervised methods: Distant-supervised methods [26], [27] generate large amounts of training data automatically by aligning unstructured text with a knowledge base. [28] attempted to incorporate distant supervision into text processing to automatically generate training samples by aligning corpus and text so as to extract a feature training classifier. [29] proposed a sentence-level model that selects valid instances and makes full use of the supervision information from a knowledge base.

5) Deep learning methods: Deep learning methods have proven to be exceptionally powerful in NLP (natural language processing) and graph identification, which has inspired their use for solving relation extraction problems. The architectures of deep networks come in many flavors. There are RNNs (recurrent neural networks [30], CNNs (convolutional neural networks) [31], combined CNNs and RNNs [32], [33], and LSTMs (long short-term memories) [34].

A summary of the different relation extraction methods used by the Contest winners follows.

- **Team UWA** mapped entities to pairs by extracting relation words like verbs, prepositions, and postpositions at the sentence level, and then combining each relation

phrase with its left and right entities to form triplets. A graph constructed from an article is used to find relations distributed across multiple sentences and the triplets are filtered by removing the entities with stop words. To visualize the graph, they display the relation names through a pre-trained attention-based Bi-LSTM model [34], which classifies the relations into groups of fixed types through a Semeval task [35].

- **Team Tmail** approached relation extraction with the Stanford OpenIE toolkit [15] and OpenIE [16]. Since they used more than one model for both named entity recognition and relation extraction tasks, they defined some hand-written rules to reduce the number of redundant triplets – for example, removing the stop words in an entity phrase (*e.g.,* 'an', 'the', 'it'), or locating noun chunks with SpaCy [13] and merging entities with the same chunks.
- **Team BUPT-IBL** primarily used the Stanford OpenIE tool [15], and designed a model based on a syntax-tree to extract more triplets, which significantly enhances the performance of their model.
- **Team MIDAS-IIITD** designed hand-written rules to derive triplets based on the POS tags of entity chunks.
- **Team Lab1105** used SpaCy [13] and designed a series of hand-written rules based on subjects, objects, predicates, and prepositions to extract triplets.

C. Co-reference Resolution

Co-reference resolution or entity resolution is used when an entity in a knowledge base is linked to multiple entity references. For instance, "President Trump" and "Donald John Trump" are the same person, so these two entity references should be merged before they are connected to an entity in the knowledge base.

In recent years, most solutions to entity resolution have been based on state-of-the-art machine learning methods. [36], [37] cast entity resolution as a classification problem and solve it with a decision tree algorithm. [38]–[40] cast entity resolution as a clustering problem and train a classifier to identify duplicate pairs.

Term similarity [41] and query context similarity [42] have both been proposed as options for overcoming some of the challenges with data sparsity and the difficulties with establishing associations between entities in different contexts, which is crucial with machine learning techniques.

Four of the five winning teams in this Contest (UWA, BUPT-IBL, MIDAS-IIITD, and Lab1105) used NeuralCoref [43] for entity resolution.

IV. Showcases

A. Awards

The 5 winning teams are listed in Table I.

TABLE I
Top 5 teams of 2019 ICDM/ICBK Contest

Team	Award	Organization
UWA	First Prize	University of Western Australia, Australia
Tmail	Second Prize	Tencent Medical AI Lab, China
BUPT-IBL	Honorable Mention	Beijing University of Posts and Telecommunications, China
MIDAS-IIITD	Honorable Mention	Indraprastha Institute of Information Technology Delhi, India
Lab1105	Honorable Mention	Wuhan University of Technology, China

B. Grading Rules and Scores

In the first stage, the triplets from each submission file were split by industry, and then compared with those triplets that were labeled by the experts (in 120 of the 300 articles in the dataset). The scores were calculated by averaging the score for each industry. The graph generated by the triplets from each text file was compared with the graph edit distance from the graph labeled by the two industry experts. We used *NetworkX* [44] as the measurement of graph edit distance. For a fair comparison, the entity words in team submissions were substituted with words from a synonym dictionary labeled by the industry experts. The best possible score was 0 – if a submission was identical to the experts' labeled triplets. The score for an empty file was around 17.51. Eight teams were selected for the next stage, including Team UWA, Tmail, MIDAS-IIITD, BUPT-IBL, Lab1105, MoFIT, SAPL, and SID.

In the second stage, the eight teams were each asked to develop a web application that takes a short piece of text as the input and outputs a knowledge graph. The submitted web applications were graded by a group of 12 scholars. From 100 marks in total, 60 marks were allocated to the quality of the graphs constructed from 10 randomly selected texts from the original dataset, 20 marks were awarded to web design, and the remaining 20 marks for the robustness of each website.

C. Showcases

We now show the knowledge graph generated by each winning team on the example text[2]:

"BYD debuted its E-SEED GT concept car and Song Pro SUV alongside its all-new e-series models at the Shanghai International Automobile Industry Exhibition. The company also showcased its latest Dynasty series of vehicles, which were recently unveiled at the company's spring product launch in Beijing."

- **Team UWA**

Figure 1 is Team UWA's visualization of the example text from their web application[3]. Their model successfully extracted all the entities and relations. It correctly recognized 'The company' in the second sentence, and 'BYD' in the second sentence as the mention of the same entity. But it did not link 'BYD' with the second 'the company' in the second sentence together as the same entity, which may be caused by the increased distance between the two phrases.

- **Team Tmail**

[2]https://www.greencarcongress.com/2019/04/20190418-byd.html
[3]https://github.com/Michael-Stewart-Webdev/text2kg-visualisation

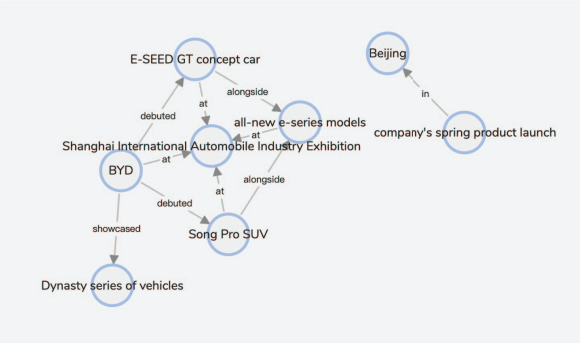

Fig. 1. Team UWA's Knowledge Graph on Example Text

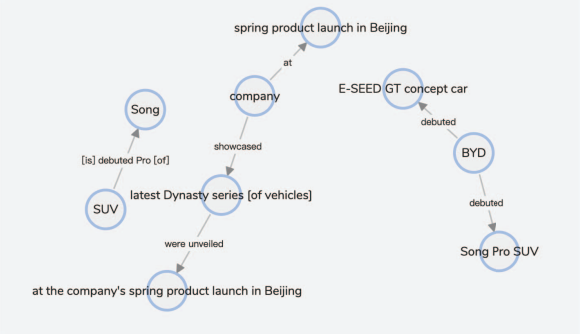

Fig. 2. Team Tmail's Knowledge Graph on Example Text

Figure 2 shows the graph produced by Team Tmail's entry. Most of the entities are extracted and linked correctly in their solution. But their model did not fully filter the duplicate triplets (*e.g.,* 'Song' and 'Song Pro SUV') and, again, it did not recognize phrases referring to the same entity, *e.g.,* 'BYD' and 'the company'.

- **BUPT-IBL**

Team BUPT-IBL's result is shown in Figure 3. Most of the entities are recognized and the co-reference resolution function performs well. Some of the triplets were not correctly derived from the original text, *e.g.,* (*E-SEED GT concept car, debuted, BYDs spring product launch*).

- **MIDAS-IIITD**

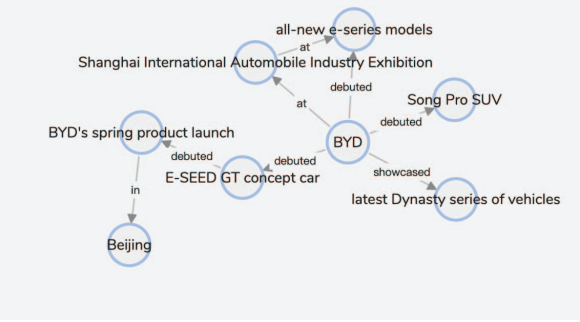

Fig. 3. Team BUPT-IBL's Knowledge Graph on Example Text

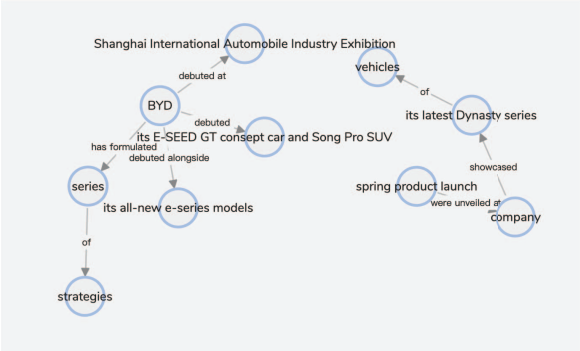

Fig. 4. Team MIDAS-IIITD's Knowledge Graph on Example Text

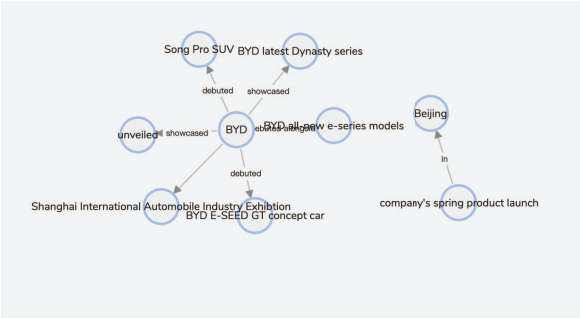

Fig. 5. Team Lab1105's Knowledge Graph on Example Text

Figure 4 shows the knowledge graph produced by Team MIDAS-IIITD. The model successfully extracted some entities and relations, but it failed to link 'company' with 'BYD'.

- **Lab1105**

Figure 5 shows Team Lab1105's knowledge graph. Most of the entities were extracted and correctly linked with each other. This team applied co-reference resolution twice, before and after entity recognition, which replaced possessive pronouns with entity terms and resulted in adding redundant terms to some entities. In addition, this model can be further improved to link 'BYD' with 'the company' in the text.

V. Conclusions

This article reported the outcomes of the 2019 ICDM/ICBK Knowledge Graph Construction Contest organized by Mininglamp Academy of Sciences and the Hefei University of Technology. Entrants were given a dataset of 300 short texts spanning four different industry domains and asked to design a method of building a knowledge graph from unstructured text without human intervention.

The Contest was divided into two rounds. In the first round, each participating team designed a model to extract knowledge graphs from the provided dataset of text articles. Competitors who qualified in this round moved onto the second round, and were requested to develop a web application to visualize knowledge graphs from the given data.

The first prize in the Contest was awarded to Team UWA, who designed a pipeline-style model. Entities were extracted

with SpaCy [13], while POS tags were identified and noun and verb phrases were chunked with predefined rules. To extract relations, they mapped the entities into pairs by extracting verbs, prepositions, and post-positions as relation words. They supplemented this process with a pre-trained attention-based Bi-LSTM model [34] to classify the relations into predefined types during the visualization step. The website they designed to visualize the triplets was also quite impressive. In addition to the basic functions, this team used centrality of nodes to zoom out important entities. Multiple documents could be fed in as inputs, and their web application displayed color-coded entities to indicate co-occurring entities between documents.

ACKNOWLEDGEMENTS

We would like to thank Mininglamp Technology (Beijing, China) for preparing labeled datasets and the contest website, the Contest committee for the evaluations of submissions, and all teams for their participation.

REFERENCES

[1] C. Xiong, R. Power, and J. Callan, "Explicit semantic ranking for academic search via knowledge graph embedding," in *Proc. WWW*, 2017, pp. 1271–1279.

[2] Z. Sun, J. Yang, J. Zhang, A. Bozzon, L.-K. Huang, and C. Xu, "Recurrent knowledge graph embedding for effective recommendation," in *Proc. ACM RecSys*, 2018, pp. 297–305.

[3] Y. Zhang, H. Dai, Z. Kozareva, A. J. Smola, and L. Song, "Variational reasoning for question answering with knowledge graph," in *Proc. AAAI*, 2018.

[4] D. Vrandečić and M. Krötzsch, "Wikidata: a free collaborative knowledge base," *Communications of the Acm*, vol. 57, no. 10, pp. 78–85, 2014.

[5] Q. Liu, Y. Li, H. Duan, Y. Liu, and Z. Qin, "Knowledge graph construction techniques," *Journal of Computer Research and Development*, vol. 53, no. 3, pp. 582–600, 2016.

[6] H. Paulheim, "Knowledge graph refinement: A survey of approaches and evaluation methods," *Semantic web*, vol. 8, no. 3, pp. 489–508, 2017.

[7] I. Augenstein, M. Das, S. Riedel, L. Vikraman, and A. McCallum, "Semeval 2017 task 10: Scienceie-extracting keyphrases and relations from scientific publications," *arXiv preprint arXiv:1704.02853*, 2017.

[8] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes*, vol. 30, no. 1, pp. 3–26, 2007.

[9] L. Liu and D. Wang, "A review on named entity recognition," *Journal of the China Society for Scientific and Technical Information*, vol. 37, no. 3, p. 329, 2018.

[10] C. Cherry and H. Guo, "The unreasonable effectiveness of word representations for twitter named entity recognition," in *Proc. NAACL*, 2015, pp. 735–745.

[11] F. Godin, B. Vandersmissen, W. De Neve, and R. Van de Walle, "Multimedia lab@ acl wnut ner shared task: Named entity recognition for twitter microposts using distributed word representations," in *Proc. EMNLP-WNUT*, 2015, pp. 146–153.

[12] R. Arora, C. Tsai, K. Tsereteli, P. Kambadur, and Y. Yang, "A semi-markov structured support vector machine model for high-precision named entity recognition," in *Proc. ACL*, 2019, pp. 5862–5866.

[13] M. Honnibal and I. Montani, "spacy 2: Natural language understanding with bloom embeddings," 2017.

[14] M. Stewart, M. Enkhsaikhan, and W. Liu, "Icdm 2019 knowledge graph contest: Team uwa," in *Proc. ICDM*, 2019.

[15] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit," in *Proc. ACL*, 2014, pp. 55–60.

[16] S. Saha and M. Mausam, "Open information extraction from conjunctive sentences," in *Proc. COLING*, 2018, pp. 2288–2299.

[17] P. Lu, T. Bai, and P. Langlais, "Sc-lstm: Learning task-specific representations in multi-task learning for sequence labeling," in *Proc. NAACL*, 2019, pp. 2396–2406.

[18] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[19] A. Akbik, D. Blythe, and R. Vollgraf, "Contextual string embeddings for sequence labeling," in *Proc. COLING*, 2018, pp. 1638–1649.

[20] E. F. Tjong Kim Sang and F. De Meulder, "Introduction to the conll-2003 shared task: Language-independent named entity recognition," in *Proc. CoNLL*, 2003, pp. 142–147.

[21] D. Xie and Q. Chang, "Review of relation extraction," *Application Research of Computers*, vol. 37, no. 7, pp. 1–5, 2019.

[22] F. Wu and D. S. Weld, "Open information extraction using wikipedia," in *Proc. ACL*, 2010, pp. 118–127.

[23] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, "Open information extraction from the web," in *Proc. Ijcai*, vol. 7, 2007, pp. 2670–2676.

[24] A. Fader, S. Soderland, and O. Etzioni, "Identifying relations for open information extraction," in *Proc. EMNLP*, 2011, pp. 1535–1545.

[25] M. Schmitz, R. Bart, S. Soderland, O. Etzioni *et al.*, "Open language learning for information extraction," in *Proc. EMNLP-CoNLL*, 2012, pp. 523–534.

[26] M. Surdeanu, J. Tibshirani, R. Nallapati, and C. D. Manning, "Multi-instance multi-label learning for relation extraction," in *Proc. EMNLP-CoNLL*, 2012, pp. 455–465.

[27] C. Quirk and H. Poon, "Distant supervision for relation extraction beyond the sentence boundary," *arXiv preprint arXiv:1609.04873*, 2016.

[28] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proc. ACL-IJCNLP*, 2009, pp. 1003–1011.

[29] G. Ji, K. Liu, S. He, and J. Zhao, "Distant supervision for relation extraction with sentence-level attention and entity descriptions," in *Proc. AAAI*, 2017.

[30] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng, "Semantic compositionality through recursive matrix-vector spaces," in *Proc. EMNLP-CoNLL*, 2012, pp. 1201–1211.

[31] Y. Lin, S. Shen, Z. Liu, H. Luan, and M. Sun, "Neural relation extraction with selective attention over instances," in *Proc. ACL*, 2016, pp. 2124–2133.

[32] X. Guo, H. Zhang, H. Yang, L. Xu, and Z. Ye, "A single attention-based combination of cnn and rnn for relation classification," *IEEE Access*, vol. 7, pp. 12 467–12 475, 2019.

[33] V.-H. Tran, V.-T. Phi, H. Shindo, and Y. Matsumoto, "Relation classification using segment-level attention-based CNN and dependency-based RNN," in *Proc. NAACL*, 2019, pp. 2793–2798.

[34] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proc. ACL*, 2016.

[35] I. Hendrickx, S. N. Kim, Z. Kozareva, P. Nakov, D. Ó Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, and S. Szpakowicz, "Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals," in *Proc. ACL*, 2009, pp. 94–99.

[36] J. F. McCarthy and W. G. Lehnert, "Using decision trees for coreference resolution," *arXiv preprint cmp-lg/9505043*, 1995.

[37] D. Bean and E. Riloff, "Unsupervised learning of contextual role knowledge for coreference resolution," in *Proc. HLT-NAACL*, 2004, pp. 297–304.

[38] M. Bilenko and R. J. Mooney, "Adaptive duplicate detection using learnable string similarity measures," in *Proc. KDD*, 2003, pp. 39–48.

[39] P. Christen, "Febrl: a freely available record linkage system with a graphical user interface," in *Proc. HDKM*, 2008, pp. 17–25.

[40] T. Cheng, H. W. Lauw, and S. Paparizos, "Entity synonyms for structured web search," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 10, pp. 1862–1875, 2011.

[41] P. Pantel, E. Crestan, A. Borkovsky, A.-M. Popescu, and V. Vyas, "Web-scale distributional similarity and entity set expansion," in *Proc. EMNLP*, 2009, pp. 938–947.

[42] K. Chakrabarti, S. Chaudhuri, T. Cheng, and D. Xin, "A framework for robust discovery of entity synonyms," in *Proc. KDD*, 2012, pp. 1384–1392.

[43] T. Wolf, "Neuralcoref 4.0: Coreference resolution in spacy with neural networks." 2017.

[44] D. A. S. Aric A. Hagberg and P. J. Swart, "Exploring network structure, dynamics, and function using networkx," in *Proc. SciPy*, 2008.