CN-DBpedia2: An Extraction and Verification Framework for Enriching Chinese Encyclopedia Knowledge Base postprint

Authors: Xu, Bo, Liang, Jiaqing, Xie, Chenhao, Liang, Bin, Chen, Lihan, Xiao, Yanghua, Xiao, Yanghua

Date: 2022-11-27T00:00:00+00:00

Abstract

Knowledge base plays an important role in machine understanding and has been widely used in various applications, such as search engine, recommendation system and question answering. However, most knowledge bases are incomplete, which can cause many downstream applications to perform poorly because they cannot find the corresponding facts in the knowledge bases. In this paper, we propose an extraction and verification framework to enrich the knowledge bases. Specifically, based on the existing knowledge base, we first extract new facts from the description texts of entities. But not all newly-formed facts can be added directly to the knowledge base because the errors might be involved by the extraction. Then we propose a novel crowd-sourcing based verification step to verify the candidate facts. Finally, we apply this framework to the existing knowledge base CN-DBpedia and construct a new version of knowledge base CN-DBpedia2, which additionally contains the high confidence facts extracted from the description texts of entities.

Full Text

Preamble

CN-DBpedia2: An Extraction and Verification Framework for Enriching Chinese Encyclopedia Knowledge Base

Bo Xu[1], Jiaqing Liang[2], Chenhao Xie[2], Bin Liang[2], Lihan Chen[2] & Yanghua Xiao[2]†
[1]School of Computer Science and Technology, Donghua University, Shanghai 200051, China
[2]Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University, Shanghai 200433, China

Keywords: Knowledge graph; Entity typing; Slot filling; Information extraction; Crowdsourcing

Citation: B. Xu, J. Liang, C. Xie, B. Liang, L. Chen & Y. Xiao. CN-DBpedia2: An extraction and verification framework for enriching Chinese encyclopedia knowledge base. *Data Intelligence* 1(2019), 244-261. doi: 10.1162/dint_a_{00017}

Received: January 4, 2019; **Revised:** May 14, 2019; **Accepted:** May 24, 2019

Abstract

Knowledge bases play a crucial role in machine understanding and have been widely used in various applications such as search engines, recommendation systems, and question answering. However, most knowledge bases are incomplete, which can cause many downstream applications to perform poorly because they cannot find the corresponding facts. In this paper, we propose an extraction and verification framework to enrich knowledge bases. Specifically, based on an existing knowledge base, we first extract new facts from entity description texts. However, not all newly extracted facts can be added directly to the knowledge base because extraction may introduce errors. We therefore propose a novel crowdsourcing-based verification step to validate candidate facts. Finally, we apply this framework to the existing CN-DBpedia knowledge base and construct a new version, CN-DBpedia2, which additionally contains high-confidence facts extracted from entity description texts.

1. Introduction

In recent years, significant efforts have been devoted to harvesting knowledge from the Web, resulting in the construction of various knowledge graphs (KGs) and knowledge bases (KBs) such as YAGO [**?**], DBpedia [**?**], Freebase [**?**], and CN-DBpedia [**?**]. These knowledge bases play important roles in many applications, including search engines [**?**], recommendation systems [**?**], and question answering [**?**]. †Corresponding author: Yanghua Xiao (Email: shawyh@fudan.edu.cn; ORCID: 0000-0001-8403-9591).

However, knowledge bases are generally incomplete. Facts in current KBs (e.g., DBpedia [**?**], YAGO [**?**], Freebase [**?**], and CN-DBpedia [**?**]) are mainly obtained from carefully edited structured texts (e.g., infobox and category information) on online encyclopedia websites (e.g., Wikipedia and Baidu Baike[1]). Since knowledge is abundant but editors have limited capacity, many structured texts are often incomplete, making the facts directly extracted from them incomplete as well. According to Catriple [**?**], only 44.2% of articles in Wikipedia have infobox information. Similarly, in Baidu Baike, the largest Chinese online

[1]http://baike.baidu.com/

encyclopedia, nearly 32% (almost 3 million) of entities lack both infobox and category information [**?**].

An incomplete knowledge base leads to poor performance in many downstream applications because they cannot find the corresponding facts. For example, if a knowledge graph lacks Donald Trump' s birthday, it cannot answer the question "when was Donald Trump born?"

To address this challenge, we propose an extraction and verification framework to enrich knowledge bases. Based on existing knowledge bases, we first extract new facts from entity description texts. However, not all newly extracted facts can be added directly to the knowledge base because errors may be introduced during extraction [**?**, **?**, **?**]. For example, Table 1 shows the F1-scores of state-of-the-art text-based extractors on the slot filling benchmark TAC dataset (development set: data from 2012/2013, evaluation set: data from 2014) [**?**], including pattern-based methods (PATdist [**?**]), traditional machine learning methods (Mintz++ [**?**], SVMskip [**?**]), graphical models (MIMLRE [**?**]), and neural network-based methods (CNNcontext) [**?**]. The facts extracted by these methods still contain significant noise. This motivates us to employ a novel crowdsourcing method to verify extracted facts. Considering human cost, we only verify low-confidence facts. Ultimately, only two types of extracted facts are added to the knowledge base: high-confidence facts and low-confidence facts that have been verified as correct by humans.

The missing facts in knowledge bases mainly include relationships between entities and relationships between entities and concepts. In this paper, we use entity description texts to enrich the knowledge base through two subtasks: entity typing and slot filling. Our contributions are as follows:

1. For the entity typing subtask, we propose a multi-instance learning model that processes both textual and heterogeneous information.
2. For the slot filling subtask, we use a transfer learning strategy to extract values for long-tailed predicates.
3. We propose a novel implicit crowdsourcing approach to verify low-confidence new facts.
4. We apply this framework to the existing CN-DBpedia knowledge base and release a new version, CN-DBpedia2, which additionally contains facts extracted from entity description texts. As of April 2019, CN-DBpedia2 contained approximately 16,024,656 entities and 228,499,155 facts.

The rest of this paper is organized as follows. Section 2 introduces the system architecture of CN-DBpedia2. Sections 3 and 4 detail the methods for entity typing and slot filling, respectively. Section 5 describes how we verify low-confidence new facts. Section 6 presents statistics for our new system. Finally, Section 7 concludes the paper.

2. System Architecture

The system architecture of CN-DBpedia2 is shown in Figure 1 [Figure 1: see original paper], which extends CN-DBpedia. CN-DBpedia2 uses Baidu Baike, Hudong Baike, Chinese Wikipedia, and other domain-specific encyclopedia websites as data sources. The pipeline consists of five components:

The **extraction component** extracts raw facts from encyclopedia articles by crawling web pages for all entities in the data sources and extracting raw facts from the structured text on those pages.

The **normalization component** normalizes raw facts, including the normalization of attributes/predicates and fact values.

The **enrichment component** extracts new facts that cannot be obtained directly from the structured text of web pages.

The **correction component** corrects erroneous facts in the knowledge base through error detection and crowdsourcing-based error correction.

The **update component** maintains the freshness of the knowledge base through periodic and active updates.

CN-DBpedia2 differs from CN-DBpedia primarily in its enrichment component. We propose an extraction and verification framework to enrich knowledge bases, which includes three new features: entity typing, slot filling, and fact verification. As shown in Figure 2 [Figure 2: see original paper], we use both entity typing and slot filling methods to extract new facts from entity description texts, and low-confidence facts must be verified before being added to the knowledge base.

3. Entity Typing

The entity typing task aims to find a set of types/concepts for each entity in a knowledge base. An entity contains both structured and unstructured features. In CN-DBpedia, we used structured features to type entities [**?**]. In CN-DBpedia2, we first use unstructured features alone to type entities and then use both structured and unstructured features together.

3.1 Entity Typing from Unstructured Features

We propose a multi-instance method called METIC [**?**] to type entities using only unstructured features. An entity may have multiple mentions in a corpus; we treat each mention of an entity in the knowledge base as an instance of that entity and learn the entity' s types from multiple instances. Specifically, we first use an end-to-end neural network model to type each entity mention (mention typing), and then use an integer linear programming (ILP) method to aggregate the predicted type results from multiple instances (type fusion). Our solution framework is shown in Figure 3 [Figure 3: see original paper].

In the offline phase, we train models for the two subtasks—mention typing and type fusion—separately. For mention typing, we model it as a multi-label classification problem. In our setting, we use distant supervision to automatically construct training data and build a supervised learning model (specifically, a neural network model). For type fusion, we model it as a constrained optimization problem and propose an integer linear programming model to solve it. The constraints are derived from semantic relationships between types.

In the online phase, we use the models built offline to enrich types for each entity in the knowledge base. For each entity e, we first employ existing entity linking systems [**?**] to discover entity mentions from its corpus. Each mention and its corresponding context $\langle m_i, c_i \rangle$ are fed into the mention typing model, which derives a set of types with each type associated with a probability (i.e., $P(t|m_i)$). Types and their probabilities derived from each mention are further fed into the integer linear programming model with constraints specifying exclusivity or compatibility among types. The model finally selects a subset from all candidate types as the final output types.

In the mention typing step, we propose a neural network model, as shown in Figure 4 [Figure 4: see original paper]. We first divide the sentence into three parts: the left context of the mention, the mention part, and the right context of the mention. Each part is fed into a parallel neural network with a similar structure: a word embedding layer and a BiLSTM (bidirectional long short-term memory) layer [**?**]. We then concatenate the outputs of these BiLSTM layers to generate the final output.

In the type fusion step, we propose an integer linear programming (ILP) model to aggregate all types derived from an entity' s mentions to reduce noise. ILP is an optimization model with constraints where all variables are required to be non-negative integers [**?**]. For each entity e, we first define a binary decision variable $x_{e,t}$ for every candidate type t, indicating whether entity e belongs to type t. Our ILP model is as follows:

$$\text{Maximize} \quad \sum_{t \in T_e} \max_{m \in M_e} P(t|m) \cdot x_{e,t} \tag{1}$$

$$\text{Subject to} \quad x_{e,t} \in \{0, 1\} \tag{2}$$

$$\sum_{t \in T_e} x_{e,t} \geq 1 \tag{3}$$

where $\max_{m \in M_e} P(t|m)$ represents the maximum probability that any mention of entity e belongs to type t, and h is the threshold (set to 0.5 in our experiments).

We propose two constraints for our ILP model: a type disjointness constraint and a type hierarchy constraint. The type disjointness constraint states that an entity cannot simultaneously belong to two semantically mutually exclusive

types, such as Person and Organization. Types with no entity overlap or insignificant overlap below a specified threshold are considered disjoint [**?**]. The type hierarchy constraint states that if an entity does not belong to type t, it certainly does not belong to any of t' s subtypes. For example, an entity that does not belong to type Artist should not be classified as type Actor.

3.2 Entity Typing from Heterogeneous Features

To leverage both structured and unstructured features of entities, we propose a new framework called METIH (Multi-instance Entity TypIng from Heterogeneous features), a modified version of METIC. As shown in Figure 5 [Figure 5: see original paper], most components are the same as METIC except those marked by dotted lines. Specifically, in the type fusion step, we treat the prediction result from structured features as a new instance of an entity and use a new ILP model to aggregate these prediction results. The new ILP model is as follows:

$$\text{Maximize} \quad \sum_{t \in T_e} \max(\max_{m \in M_e} P(t|m), d(t \in C_e)) \cdot x_{e,t} \tag{4}$$

$$\text{Subject to} \quad x_{e,t} \in \{0, 1\} \tag{5}$$

$$\sum_{t \in T_e} x_{e,t} \geq 1 \tag{6}$$

$$\text{ISA}(t_i, t_j) \cdot x_{e,t_i} \leq x_{e,t_j} \tag{7}$$

where function $d(t \in C_e)$ is defined as follows: if type t belongs to C_e (the set of types from structured features), then $d(t \in C_e) = 1$; otherwise, $d(t \in C_e) = 0$. Thus, $\max(\max_{m \in M_e} P(t|m), d(t \in C_e))$ represents the maximum probability that entity e belongs to type t, where $\max_{m \in M_e} P(t|m)$ is the maximum probability from unstructured features and $d(t \in C_e)$ is the probability from structured features.

4. Slot Filling

Some entities' attribute values cannot be directly extracted due to missing information, so we extract these values from text corpora. We cast this as a slot filling task: given an entity and an attribute, our goal is to extract values from the entity' s description text. State-of-the-art methods typically use supervised learning to build extractors for each predicate, treating it as a sequence tagging problem.

However, training samples for each predicate are unbalanced: head predicates usually contain large amounts of training data while long-tailed predicates have only a few samples. In CN-DBpedia, we extracted values for head predicates. In CN-DBpedia2, we focus on extracting values for long-tailed predicates.

A naive solution for slot filling is to train a separate predicate extractor for each predicate. Single predicate extractors are trained separately on data for different predicates. For long-tailed predicates with insufficient training data, these extractors may not be fully trained, impairing performance. Therefore, we propose a Multiple Predicate Extractor with Prior Knowledge (MPK) model. The MPK model also takes the predicate as input. We first pre-train a model using all head predicates and then fine-tune it for each long-tailed predicate through transfer learning. Since most parameters in the model are shared across different predicates, long-tailed predicates can leverage abundant training data from head predicates via transfer learning.

Naturally, we want to utilize training samples from other predicates, motivating us to develop a multiple predicate extractor structure. The network structure is shown in Figure 6 [Figure 6: see original paper]. The extractor consists of five parts: (1) text embedding layer, (2) knowledge embedding layer, (3) text-knowledge attention layer, (4) encoder layer, and (5) output layer.

4.1 Text Embedding Layer

We first pre-process sentences. The text embedding layer captures text features before extraction, including both word and phrase information. We then encode the concatenation of these embedded vectors with a BiLSTM layer.

4.2 Knowledge Embedding Layer

This layer embeds prior knowledge as guidance for information extraction, determining what kind of information should be extracted. The prior knowledge includes the type of the entity (subject) and the predicate specifying what information to extract.

Type Encoder Layer. One entity may belong to multiple types, implying different aspects of the entity. We use a self-attention layer to embed all types of an entity, employing the multi-head attention mechanism proposed by [**?**]. Specifically, for each type (the query), we compute a weighted sum of all types (keys) in the input based on similarity between query and key measured by dot product.

Predicate-Type Attention Layer. We combine type and predicate information in this layer. The predicate and type together determine what to extract. For example, with entity type Person and predicate birthplace, we can determine the task is to extract where the person was born, similar to a query. However, an entity often has many noisy types (e.g., actor and producer in Figure 6). We use predicate-to-type attention to select suitable types for the query. Specifically, we use T and P to denote encoded types and predicate, respectively. The predicate-to-type similarity is computed as $S_0 \in \mathbb{R}^{1\times|T|}$. We normalize the only row of S_0 by applying the softmax function, deriving A_0. The similarity function used here is the trilinear function [**?**]: $f(t,p) = W_0[t, p, t \odot p]$, where $\odot$ is element-wise multiplication and W_0 is a trainable variable.

The predicate-to-type attention is computed as $U = A_0^T \cdot T$. The output is an encoded knowledge sequence U, where each unit is $u = W_1[t, p, t \odot p, a_0^t]$, with t and p denoting an encoded type and predicate embedding, respectively, and a_0 is an attention value in A_0. Each unit u specifies an aspect of the extraction task, such as the birthPlace of a person.

4.3 Text-Knowledge Attention Layer

This module uses knowledge as a query to guide the extractor. We use H and U to denote encoded text and knowledge, respectively. We adopt attention similar to BiDAF [**?**] to model interaction between text and knowledge, including text-to-knowledge attention and knowledge-to-text attention.

Text-to-knowledge attention is constructed as follows. We first compute similarities between each pair of encoded text and knowledge using Equation (2), rendering a similarity matrix S. We then normalize each row of S by applying the softmax function, obtaining matrix A. The text-to-knowledge attention is computed as $A = \text{softmax}(S) \cdot U$.

We additionally use knowledge-to-text attention. Specifically, we perform max pooling on each column of S and compute the softmax function to get the knowledge-to-text attention $B = \text{softmax}(\max_{\text{col}}(S))$.

4.4 Model Encoder Layer

The input to this layer at each position is $[h, a, h \odot a, h \odot b]$, where a and b are respectively a row of attention matrices A and B. This module contains two BiLSTM layers.

4.5 Output Layer

In the output layer, we adopt a Conditional Random Field layer, which has proven effective for sequence tagging tasks [**?**].

4.6 Training

To address insufficient training data for long-tailed predicates, we enable the model to benefit from abundant training samples of head predicates. Specifically, we let the model first learn to extract values for head predicates and then learn extraction for long-tailed predicates through transfer learning.

We build extractors for each long-tailed predicate. The training process includes two steps: first, we pre-train the MPK model on training data from the top K head predicates; then we fine-tune the MPK model on training data for the long-tailed predicate.

5. Fact Verification

Through entity typing and slot filling, we can obtain many facts from text, but some are incorrect. Adding these errors to an existing knowledge base would reduce its quality. Hence, we need to verify low-confidence facts through human judgment before adding them to the knowledge base. The verification task is as follows: given a text passage and a fact extracted from it, we ask people to judge whether the fact can be inferred from the text.

Verifying all facts by experts is prohibitively expensive. To solve this problem, we propose a novel implicit crowdsourcing approach that asks users to verify extracted facts. Our idea is inspired by CAPTCHA [**?**], a program that generates and grades tests that most humans can pass but current computer programs cannot, thereby differentiating humans from computers.

As shown in Figure 7 [Figure 7: see original paper], CAPTCHA asks users to type distorted text seen in an image. If the typed characters match those in the image, the user is considered human.

We propose a different type of CAPTCHA called rcCAPTCHA, a reading comprehension test. Our goal is to have people verify whether a fact can be inferred from a text passage. For each test, the reading passage is a text snippet, the question is generated from the fact, and answer options are generated from both the text and the fact.

We propose different ways to generate questions and answer options for different fact types. For a fact derived from entity typing, such as (e, isA, t) where $e \in E$ is an entity and $t \in T$ is a type, we generate a question like "Which type does $< e >$ belong to?" The answer options are t and all its siblings in the type taxonomy, with t being the correct answer. Since one entity may belong to many types, entity e might belong to some sibling types of t according to other extracted facts; we exclude those types from the answer options. For a fact derived from slot filling, such as (e, a, v) where $a \in A$ is an attribute and v is the attribute value, we generate a question like "What is the $< a >$ of $< e >$?" The answer options are all words in the text (including v), with v being the correct answer. For each extracted fact, we generate several reading comprehension tests. When multiple users click the correct answer and pass the test, the extracted fact is considered correct.

To verify the accuracy of our extracted facts, we released a free rcCAPTCHA API[2] and deployed it on multiple websites. The CN-DBpedia search engine[3] is one such website; our rcCAPTCHA system is triggered when user searches exceed a certain threshold. Search can only continue if the user correctly answers the rcCAPTCHA question. As shown in Figure 8 [Figure 8: see original paper], this instance verifies whether the fact (Galare Thong Tower, star rating, 3-star) is correct. Based on this fact, the system generates the question "What is the

[2]http://kw.fudan.edu.cn/apis/supervcode/
[3]http://kw.fudan.edu.cn/cndbpedia/

star rating of Galare Thong Tower?" and provides a text containing the answer ("Galare Thong Tower is a 3-star hotel located in Chiang Mai···It is a 9-minute drive from Wat Chiang Man···"). If the majority of users click "3-star" in the text, the triple is considered correct.

Our rcCAPTCHA system is triggered thousands of times per day on average, with each fact verified by an average of 20 users. We randomly sampled 100 extracted facts that were verified as correct and achieved 100% accuracy. Although current verification speed is slow, it will accelerate as more people use our system.

6. Statistics for CN-DBpedia2

We present CN-DBpedia2 statistics in this section. The number of entities and facts in CN-DBpedia2 is much larger than in CN-DBpedia. As of April 2019, CN-DBpedia2 contains approximately 16,024,656 entities and 228,499,155 facts, while CN-DBpedia contains only 10,341,196 entities and 88,454,264 facts. Table 2 shows fact types in CN-DBpedia2 and changes compared to CN-DBpedia. The increase comes mainly from three aspects: new data sources, use of textual information, and implementation of updates.

Using our new entity typing method, we found more types for entities in CN-DBpedia2. Table 3 shows the top 20 concepts and the number of entities they contain in CN-DBpedia2, along with changes compared to CN-DBpedia. By considering textual information, entities have gained more concepts, and concept sizes have increased significantly (except for the two concepts of companies and singles, where the increase mainly comes from extraction from domain encyclopedia websites).

Based on CN-DBpedia2, we have also published new APIs[4] for natural language understanding, including entity linking and question answering. As of April 2019, these APIs had been called 950 million times since their publication in December 2015.

7. Conclusion

In this paper, we released CN-DBpedia2, a new version of the knowledge base that extends CN-DBpedia. Based on the existing knowledge base, we additionally exploit both structured and unstructured features for entity typing and propose a transfer learning strategy to extract values for long-tailed predicates. We also propose a novel implicit crowdsourcing approach to verify low-confidence new facts, adding only those verified as correct to the knowledge base. As of April 2019, CN-DBpedia2 contained approximately 16,024,656 entities and 228,499,155 facts, and the APIs had been called 950 million times.

[4]http://kw.fudan.edu.cn/apis

Author Contributions

This work represents collaboration among all authors. Y. Xiao (shawyh@fudan.edu.cn) leads the CN-DBpedia project. B. Xu (xubo@dhu.edu.cn) led the work and summarized the entity typing component. C. Xie (redreamality@gmail.com) and L. Chen (lhc825@gmail.com) summarized the slot filling component. J. Liang (l.j.q.light@gmail.com) summarized the fact verification component. B. Liang (liangbin@fudan.edu.cn) summarized the statistics component. All authors made meaningful and valuable contributions to revising and proofreading the manuscript.

Acknowledgements

This work was supported by the National Key R&D Program of China (No. 2017YFC1201203), the Shanghai Sailing Program (No. 19YF1402300), and the Initial Research Funds for Young Teachers of Donghua University (No. 112-07-0053019).

References

[1] F.M. Suchanek, G. Kasneci, & G. Weikum. YAGO: A core of semantic knowledge. In: *Proceedings of the 16th International Conference on World Wide Web*, 2007, pp. 697–706. doi: 10.1145/1242572.1242667.

[2] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, & Z. Ives. DBpedia: A nucleus for a Web of open data. In: K. Aberer et al. (eds.) *The Semantic Web*. Berlin: Springer, 2007, pp. 722–735. doi: 10.1007/978-3-540-76298-0_{52}.

[3] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, & J. Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In: *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, 2008, pp. 1247–1250. doi: 10.1145/1376616.1376746.

[4] B. Xu, Y. Xu, J. Liang, C. Xie, B. Liang, W. Cui, & Y. Xiao. CN-DBpedia: A never-ending Chinese knowledge extraction system. In: *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, 2017, pp. 428–438. doi: 10.1007/978-3-319-60045-1_{44}.

[5] C. Xiong, R. Power, & J. Callan. Explicit semantic ranking for academic search via knowledge graph embedding. In: *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 1271–1279. doi: 10.1145/3038912.3052558.

[6] D. Yang, J. He, H. Qin, Y. Xiao, & W. Wang. A graph-based recommendation across heterogeneous domains. In: *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*, 2015, pp. 463–472. doi: 10.1145/2806416.2806523.

[7] W. Cui, Y. Xiao, & W. Wang. KBQA: An online template based question answering system over Freebase. In: *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016)*, 2016, pp. 4240–4241. Available at: https://www.ijcai.org/Proceedings/16/Papers/640.pdf.

[8] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P.N. Mendes, S. Hellmann, ···& C. Bizer. DBpedia—A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web Journal* 6(2) (2015), 167–195. doi: 10.3233/SW-140134.

[9] Q. Liu, K. Xu, L. Zhang, H. Wang, Y. Yu, & Y. Pan. Catriple: Extracting triples from Wikipedia categories. In: *Asian Semantic Web Conference*, 2008, pp. 330–344. doi: 10.1007/978-3-540-89704-0_{23}.

[10] B. Xu, Y. Zhang, J. Liang, Y. Xiao, S.-W. Hwang, & W. Wang. Cross-lingual type inference. In: *International Conference on Database Systems for Advanced Applications*, 2016, pp. 447–462. doi: 10.1007/978-3-319-32025-0_{28}.

[11] H. Adel, B. Roth, & H. Schütz. Comparing convolutional neural networks to traditional models for slot filling. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 828–838. Available at: https://www.aclweb.org/anthology/N16-1097.

[12] Y. Zhang, V. Zhong, D. Chen, G. Angeli, & C.D. Manning. Position-aware attention and supervised data improve slot filling. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 35–45. doi: 10.18653/v1/D17-1004.

[13] B. Xu, Z. Luo, L. Huang, B. Liang, Y. Xiao, D. Yang, & W. Wang. METIC: Multi-instance entity typing from corpus. In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 903–912. doi: 10.1145/3269206.3271804.

[14] M. Mintz, S. Bills, R. Snow, & D. Jurafsky. 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*, 2009, pp. 1003–1011. doi: 10.3115/1690219.1690287.

[15] M. Surdeanu, J. Tibshirani, R. Nallapati, & C.D. Manning. 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*, 2012, pp. 455–465. Available at: https://dl.acm.org/citation.cfm?id=2391003.

[16] L. Chen, J. Liang, C. Xie, & Y. Xiao. Short text entity linking with fine-grained topics. In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 457–466. doi:

10.1145/3269206.3271809.

[17] S. Hochreiter, & J. Schmidhuber. Long short-term memory. *Neural Computation* 9(8) (1997), 1735–1780. doi: 10.1162/neco.1997.9.8.1735.

[18] J. Clarke, & M. Lapata. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research* 31 (2008), 399–429. doi: 10.1613/jair.2433.

[19] N. Nakashole, T. Tylenda, & G. Weikum. Fine-grained semantic typing of emerging entities. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 2013, pp. 1488–1497. Available at: https://www.aclweb.org/anthology/P13-1146.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L.u. Kaiser, & I. Polosukhin. Attention is all you need. In: *The 31st Conference on Neural Information Processing Systems (NIPS 2017)*, 2017, pp. 5998–6008. Available at: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

[21] M.J. Seo, A. Kembhavi, A. Farhadi, & H. Hajishirzi. Bidirectional attention flow for machine comprehension. arXiv preprint. arXiv:1611.01603, 2016.

[22] N. Reimers, & I. Gurevych. Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 338–348. doi: 10.18653/v1/D17-1035.

[23] L. von Ahn, M. Blum, N.J. Hopper, & J. Langford. CAPTCHA: Using hard AI problems for security. In: *International Conference on the Theory and Applications of Cryptographic Techniques*, 2003, pp. 294–311. doi: 10.1007/3-540-39200-9_{18}.

Author Biography

Bo Xu is currently a Lecturer at the School of Computer Science and Technology, Donghua University. He received his PhD degree from Fudan University in 2018. His research interests include knowledge base construction and applications. He has published several papers at major conferences including the International Joint Conference on Artificial Intelligence (IJCAI), the Conference on Information and Knowledge Management (CIKM), the European Conference on Principles of Data Mining and Knowledge Discovery (PKDD), and the International Conference on Database Systems for Advanced Applications (DASFAA). He won the Best Student Paper Award at the 31st National Database Conference (NDBC 2014).

Jiaqing Liang received his Bachelor' s degree from the School of Computer Science, Fudan University, China, in 2015. He is currently working toward his PhD degree at the School of Computer Science, Fudan University, China. His research interests include knowledge bases and deep learning for text data.

Chenhao Xie is currently a PhD candidate at the School of Computer Science, Fudan University. He is the co-founder of Shuyan Inc (http://shuyantech.com). His main research interests include information extraction and knowledge graphs. He has published several papers at conferences including the International Joint Conference on Artificial Intelligence (IJCAI), the Conference on Information and Knowledge Management (CIKM), and the International Conference on Data Mining (ICDM).

Bin Liang is a PhD candidate at the School of Computer Science, Fudan University, China. He also completed his undergraduate studies at the Department of Computer Science and Technology, Fudan University. His research interests include data science, knowledge graphs, recommender systems, and social network mining.

Lihan Chen is a PhD candidate at the School of Computer Science, Fudan University, China. His research interests include entity linking and information extraction.

Yanghua Xiao is a Professor at the School of Computer Science, Fudan University. He is one of the young "973" scientists. His research interests include big data management and mining, graph databases, and knowledge graphs. Recently, he has published more than 70 papers in international leading journals and top conferences. He won the Best PhD Thesis Nomination of the Chinese Computer Federation (CCF) in 2010, the CCF2014 Natural Science Award (second level), and the ACM (CCF) Shanghai Distinguished Young Scientist Nomination Award.

Note: Figure translations are in progress. See original paper for figures.