CN-DBpedia: A Never-Ending Chinese Knowledge Extraction System

Bo Xu[1], Yong Xu[1], Jiaqing Liang[1,2], Chenhao Xie[1,2], Bin Liang[1], Wanyun Cui[1], and Yanghua Xiao[1,3](✉)

[1] Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University, Shanghai, China
{xubo,yongxu16,jqliang15,xiech15,liangbin,shawyh}@fudan.edu.cn, wanyuncui1@gmail.com
[2] Data Eyes Research, Shanghai, China
[3] Shanghai Internet Big Data Engineering and Technology Center, Shanghai, China

Abstract. Great efforts have been dedicated to harvesting knowledge bases from online encyclopedias. These knowledge bases play important roles in enabling machines to understand texts. However, most current knowledge bases are in English and non-English knowledge bases, especially Chinese ones, are still very rare. Many previous systems that extract knowledge from online encyclopedias, although are applicable for building a Chinese knowledge base, still suffer from two challenges. The first is that it requires great human efforts to construct an ontology and build a supervised knowledge extraction model. The second is that the update frequency of knowledge bases is very slow. To solve these challenges, we propose a never-ending Chinese Knowledge extraction system, `CN-DBpedia`, which can automatically generate a knowledge base that is of ever-increasing in size and constantly updated. Specially, we reduce the human costs by reusing the ontology of existing knowledge bases and building an end-to-end facts extraction model. We further propose a smart active update strategy to keep the freshness of our knowledge base with little human costs. The 164 million API calls of the published services justify the success of our system.

1 Introduction

With the boost of Web applications, WWW has been flooded with information on an unprecedented scale. However, most of which are readable only by human but not by machines. To make the machines understand the Web contents, great efforts have been dedicated to harvesting knowledge from online encyclopedias, such as Wikipedia. A variety of `knowledge graphs` or `knowledge bases` thus

This paper was supported by National Key Basic Research Program of China under No. 2015CB358800, by the National NSFC (No. 61472085, U1509213), by Shanghai Municipal Science and Technology Commission foundation key project under No. 15JC1400900, by Shanghai Municipal Science and Technology project under No. 16511102102.

S. Benferhat et al. (Eds.): IEA/AIE 2017, Part II, LNAI 10351, pp. 428–438, 2017.
DOI: 10.1007/978-3-319-60045-1_44

have been constructed, such as Yago [7], DBpedia [1] and Freebase [2]. These knowledge bases play important roles in many applications, such as question answering [3] and recommendation systems [10].

Nowadays, English language dominates the existing knowledge bases. Non-English knowledge bases, such as Chinese ones, are still very rare. For example, DBpedia is a community effort to extract structured, multilingual knowledge from 111 different language editions of Wikipedia. Its English version contains more than 3.7 million entities while its Chinese version contains only about one million entities [4]. The popularity of Chinese applications demands more Chinese knowledge bases.

Although there already exists many construction methods for knowledge bases, such as Yago [7], DBpedia [1] and zhishi.me [5] (a Chinese knowledge base), we still face two challenges:

1. First, *how to reduce human cost?* In general, it is not trivial to construct a knowledge base with only little human cost. Many hand-crafted knowledge bases greatly rely on human efforts. For example, DBpedia uses crowd sourcing method to construct an ontology [4], which is used for typing entities. Yago uses human specified patterns to enrich ontology [7]. Furthermore, many supervised methods are used to enrich knowledge bases, and most of them rely on hand-crafted features.
2. Second, *how to keep the freshness of knowledge bases?* The update frequency of existing knowledge bases are very low. For example, the update frequency of DBpedia is 6 months. However, emerging entities appear very fast, leading to the obsoleteness of knowledge bases. One direct solution to ensure the freshness of a knowledge base is synchronizing the knowledge base with the data source with a high frequency. However, a high synchronization frequency usually leads to a costly and wasteful update since most entities keep unchanged. Hence, we need a smart active update strategy, which can automatically identify the emerging entities and the changed facts of an entity already in the knowledge base.

To solve these problems, we propose a never-ending Chinese knowledge extraction system: `CN-DBpedia`. Our contributions are as follows:

1. First, we reduce the human cost in constructing the Chinese knowledge bases. We first propose to reuse an existing ontology to alleviate the difficulties to build a high-quality ontology. We also propose an end-to-end deep learning model without any human supervision to automatically extract structured facts from encyclopedia articles.
2. Second, we propose a smart active update strategy to keep the freshness of knowledge base with little update cost.

The rest of this paper is organized as follows. In Sect. 2, we present the architecture of `CN-DBpedia`. In Sect. 3, we introduce how we reduce the human efforts in building `CN-DBpedia`, including reusing the ontology of DBpedia and building an end-to-end extraction models to enrich the infobox of entities. In

Sect. 4, we elaborate our update strategies. Section 5 presents the statistic of our system, including the distribution of the extracted knowledge and the usage of information. Finally, we conclude our paper in Sect. 6.

2 System Architecture

The system architecture of CN-DBpedia is shown in Fig. 1. The system uses Baidu Baike, Hudong Baike and Chinese Wikipedia (which are the three largest Chinese encyclopedia websites) as the main data sources. We first extract raw knowledge from the articles of those encyclopedia websites. Different with Wikipedia, Baidu Baike and Hudong Baike do not provide any dump files. Hence, we need to use a crawler to fetch those articles. Encyclopedia articles contain many structured information, such as abstract, infobox and category information. We extract facts from those structured information directly and populate them into a Chinese knowledge base.

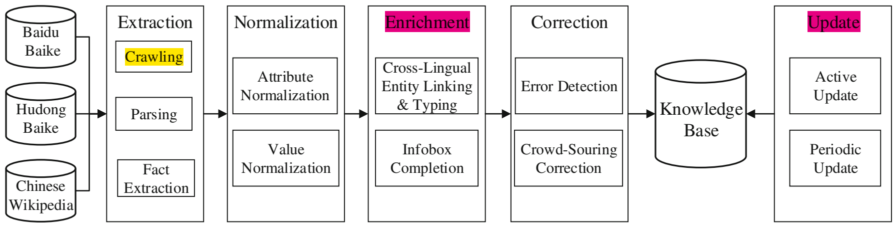
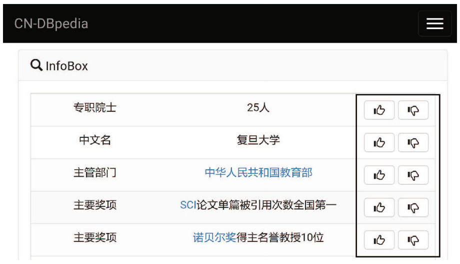


Fig. 1. System architecture of `CN-DBpedia`.

However, the quality of the knowledge base derived so far is still not good enough yet. There are three additional steps to improve the quality of the knowledge base.

- *STEP 1: Normalization.* Since online encyclopedias are user-generated content websites, the descriptions of contents are usually diverse. Different contributors may use different names and formats to describe the same attributes and values. For example, in order to describe a person's birth, one editor may use attribute `birthday`, another editor may use `date of birth` or something else. Hence, we need to normalize the attributes and values in the knowledge base.
- *STEP 2: Enrichment.* Current knowledge base is incomplete since no entities contain type information and some entities lacking of infobox information. To enrich entities with types, we reuse the ontology (especially the taxonomy of types) in DBpedia and type Chinese entities with DBpedia types. We complete the infobox information of entities by finding more SPO (Subjecte-Predicate-Object) triples from their article texts. The details are elaborated in Sect. 3.

– *STEP 3: Correction.* Current knowledge base might contain wrong facts. We propose two main steps to correct them. The first is error detection and the second is error correction. We use two methods for error detection. The first is rule-based detection. For example, we can use the domain and range of properties to find errors, the range of attribute `birthday` is date, any other types of values are wrong. The second way of error detection is based on user feedbacks. CN-DBpedia provides a user exploration interface to browse the knowledge in our system[1]. The interface allows users to provide feedbacks on the correctness of an SPO fact. The interface is shown in Fig. 2. After error detection, we mainly use crowd-sourcing for error correction. We assign error facts to different contributors and aggregate their corrections. The challenge is how to aggregate the multiple, noisy contributor inputs to create a consistent data [6]. A simple-yet-effective method is majority vote.



专职院士	25人	👍 👎
中文名	复旦大学	👍 👎
主管部门	中华人民共和国教育部	👍 👎
主要奖项	SCI论文单篇被引用次数全国第一	👍 👎
主要奖项	诺贝尔奖得主名誉教授10位	👍 👎

Fig. 2. The interface of `CN-DBpedia` allows users to provide feedbacks

3 Human Effort Reduction

Our first idea of human efforts reduction is reusing the ontology of existing knowledge bases and typing Chinese entities with the types in the existing knowledge base. The second idea is building an end-to-end extractor to complete the infobox. Next, we elaborate these two ideas.

3.1 Cross-Lingual Entity Typing

Building a widely accepted taxonomy requires huge human effort. To avoid this, we reuse the taxonomy of DBpedia because it is well defined and widely acknowledged. The first step of the taxonomy reuse is typing Chinese entities with English DBpedia types. To solve the problem, we propose a system, `CUTE` [9] (**C**ross-ling**U**al **T**ype inf**E**rence).

Figure 3 is the system architecture of `CUTE`. The system builds a supervised hierarchical classification model, which accepts an untyped Chinese entity (with

[1] http://kw.fudan.edu.cn/cndbpedia/search/.

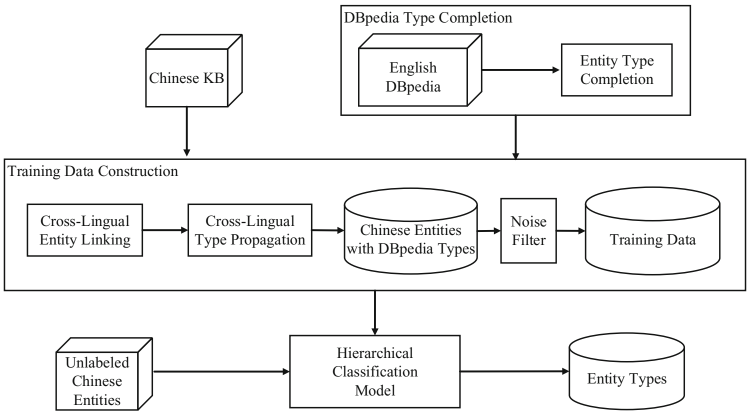

DBpedia Type Completion
Chinese KB
English DBpedia
Entity Type Completion
Training Data Construction
Cross-Lingual Entity Linking
Cross-Lingual Type Propagation
Chinese Entities with DBpedia Types
Noise Filter
Training Data
Unlabeled Chinese Entities
Hierarchical Classification Model
Entity Types

Fig. 3. Framework of `CUTE`

its features) as input and outputs all valid English types in DBpedia. Compared with the flat classification method, a hierarchical model reduces the classification time and increases the accuracy. Thus, the key to build an effective model is reduced to the construction of a high quality labeled data set. Since reducing the human efforts is our major concern, we develop an automatic labeled data generation procedure. We notice that some entities in English/Chinese knowledge bases may have the same Chinese label names. Thus, we can pair a Chinese entity with the English entity sharing the same Chinese label names. The Chinese entity as well as the types of the paired English entity is naturally a labeled sample. However, the quality of the training data constructed in the way above still has many problems:

- The types of English DBpedia entities in many cases are incomplete, which leads to the labeled samples with incomplete types.
- The types of English DBpedia entities in some cases are wrong, which leads to some samples with wrong types.
- Entity linking by Chinese label name is subject to errors, which leads to wrong samples. For example, `The Hunger Games` is a novel's label name in DBpedia, while in Chinese knowledge base, it is a label name of a film.
- The features of Chinese entities are usually incomplete. For example, some actors may only contain some basic information, such as `birthday` and `height`, which disables an effective inference to their fine-grained types, such as `Actor`.

To improve the quality of training data, we propose two approaches:

- To solve the incompleteness problem, we first complete the types for English DBpedia entities.
- To solve the last three problems, we execute a noisy filter step on training data to get rid of all wrong samples.

3.2 Infobox Completion

Infobox completion is the key step to enrich CN-DBpedia. The automation of this step is critical for the reduction of human effort. Infoboxes contain structured ⟨*subject, predicate, object*⟩ (i.e. ⟨s,p,o⟩ triple) facts about an entity, such as ⟨`Leonardo DiCaprio`, `BirthPlace`, `Hollywood`⟩. Infobox completion is a task to extract *object* for a given pair of entity and predicate from encyclopedia articles. We model the extraction problem as a seq2seq learning problem [8]. The input is a natural language sentence containing tokens, the output is the label of each token. The label is either 1 or 0, predicting whether a token is a part of the object for the predicate. We train an extractor $\mathcal{E}_p$ for each *predicate*. For example, the sentence in Fig. 4 occurs in the Wikipedia page of `Leonardo DiCaprio`. The extractor of `BirthPlace` will label `Hollywood` and `California` as `True` for `Leonardo DiCaprio`.

There are two key issues to build an effective extractor. The first is how to construct the training data. The second is how to select the desired extraction model. For the first issue, we use distant supervision method. The basic idea is that the Wikipedia infobox contains many structured facts about entities and many of these facts are also mentioned in the free text part in the entity article. Thus, we can use the sentences that express the facts in the infobox as our training data. For example, if ⟨`Leonardo DiCaprio`, `BirthPlace`, `Hollywood`⟩ occurs in the infobox of `Leonardo DiCaprio`, we can easily find the sentences (shown in Fig. 4) and correctly label the object in the sentence.

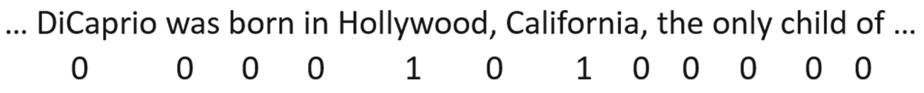

... DiCaprio was born in Hollywood, California, the only child of ...
0 0 0 0 1 0 1 0 0 0 0 0

Fig. 4. `Hollywood` and `California` extracted as *objects*.

For the second issue, we employ Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) for information extraction. LSTM-RNN has been proved effective in modeling and processing complex information, and it has achieved the state-of-the-art results in many natural language processing tasks. However, LSTM-RNN has been rarely used for information extraction. We envision that LSTM-RNN could be a powerful tool for information extraction for the following reasons. First, given labeled data, it is possible for the deep learning framework to derive the features and representation, which saves the cost of feature engineering in large scale, multiple predicates information extraction. Second, LSTM-RNN can better handle long distance dependency, and is capable of generalization of syntactic patterns of natural language.

The model structure is shown in Fig. 5. The input text is treated as a token sequence. Our final output is a labeled sequence (with a `TRUE` or `FALSE` label), which has an equal length with the input token sequence. We use a hybrid representation for each token in the input token sequences, to cover as many

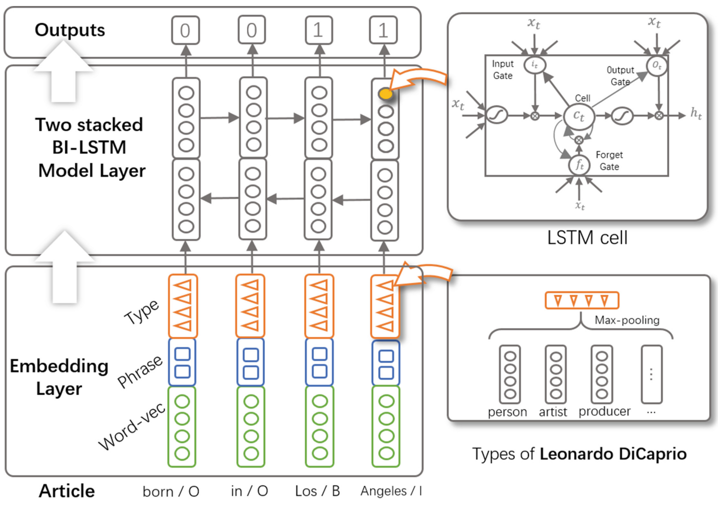


Fig. 5. Model structure [8]

as possible useful information towards knowledge base construction. The vector representation is the concatenation of three vectors: *word embedding vector (V), phrase information(P), type information (T)*. V represents the literal word information. P represents our prior knowledge about how words are combined into a phrase. T is the type representation for entity e. The representation for each token derived in the previous step is now fed into the LSTM recurrent neural network. Clearly, the label of a token in a natural language sentence is related to both its preceding tokens and its successive tokens. This motivates us to use *bi-directional hidden LSTM* layers to make use of the past and future input features.

Finally we use *binary cross entropy loss* function as the objective function to train the model:

$$Loss = \sum_{t=0}^{len} \hat{y}_t \log(y_t) + (1 - \hat{y}_t) \log(1 - y_t) \tag{1}$$

where $\hat{y}_t$ (0 or 1) is the ground-truth for the t-th token. We derive $\hat{y}_t$ in the training data generation phrase.

4 Knowledge Base Update

In this section, we elaborate our update mechanisms. We first present a widely used update strategy, then we present a more smart update strategy.

Periodical Update. The update policy is critical for the freshness of a knowledge base. The most preliminary update policy is periodical update. Existing knowledge bases such as DBpedia and Yago mostly use this update strategy. That is replacing the current knowledge bases completely with a new version after a certain period. In periodical update, the period is a key issue. A long update period means a low update cost, but the knowledge bases are likely to contain more obsolete facts about entities. A short update period can keep the freshness of the knowledge bases, but has a huge update cost. How to set a best update period is a challenging problem.

To solve this problem, CN-DBpedia supports not only periodical update but also a more smart update strategy which actively updates the knowledge bases by monitoring the changes of entities and updates an entity only when its facts change. We refer to this update strategy as *active update*, which will be elaborated in the following texts.

Active Update. The key issue of active update is to identify new entities (such as `iPhone 7s`) or an old entity (already existing in knowledge bases) that is likely to contain new facts (such as `Donald Trump`). We resort to two sources to identify these entities:

– First, entities mentioned in recent hot news.
– Second, entities mentioned in popularly searched keywords of search engines or other popular websites.

Hot news usually is about a timely and important event. The entities mentioned in the hot news are either new entities or entities whose facts tend to change. For example, Donald Trump was elected as the 45th President of the United States. His occupation in the knowledge bases should be updated to `President`. To find entities mentioned in hot news, we build a real-time news monitor, to collect the titles of latest hot news. Then, we identify the entities in the news titles and retrieve the entity from the encyclopedia source.

The popular search key words or sentences in the search engine websites usually contain the target entities. Many Chinese search engines, such as Baidu, Sogou, etc., have a panel for the real-time hot queries or topics. Moreover, some search engines such as Sogou even show the top-10 searched movies, songs, games, etc. They are also high-quality sources from which to find emerging entities or recent updated entities.

The only remaining problem is extracting entity names from news titles or search queries. To ensure our solution misses no new entities, we do not use a new phrase detection algorithm. Instead, we employ a simple greedy entity extraction method. We first do a word segmentation, to convert the sentences/phrases into a word list. Then we select all sub-lists of this list, and concatenate each sub-list into a string. In this way, we get many sub-strings. Some of them are entity names, and some others are sentence fragments. No matter what a sub-string is, we search it in the encyclopedia website. Then we can judge whether it is an entity name (there are hit results) or a meaningless fragments (no results returned). In addition, some sub-strings with low IDF (Inverse Document Frequency) could be filtered because they are usually non-meaningful entities.

5 Statistics for CN-DBpedia

We present the statistics of our system in this section. Since `CN-DBpedia` is continually updated. The entities and facts in `CN-DBpedia` are continuously growing. By December 2016, `CN-DBpedia` contains 10,341,196 entities and 88,454,264 relations. Table 1 shows the distribution of entities over different types (only top-15 most popular types are shown). We can see that our knowledge base in general covers entities from a variety of different domains. Table 2 further shows the distribution of structured facts. It is clearly to see that facts in infobox play a dominant role.

Table 1. Top-15 most popular types in CN-DBpedia.

Types	Count	Rank
dbo:Work	2,529,054	1
dbo:Agent	2,004,923	2
dbo:Person	1,217,988	3
dbo:Place	1,197,263	4
dbo:WrittenWork	1,098,019	5
dbo:Book	1,056,106	6
dbo:Organisation	790,974	7
dbo:PopulatedPlace	616,022	8
dbo:ArchitecturalStructure	492,580	9
dbo:Settlement	462,082	10
dbo:Building	454,448	11
dbo:Company	417,010	12
dbo:Species	211,536	13
dbo:Eukaryote	207,771	14
dbo:Food	178,689	15

Table 2. Relations in CN-DBpedia.

Relation types	Count	Rank
Entity Infobox	41,140,062	1
Entity Tags	19,865,811	2
Entity Types	19,846,300	3
Entity Information	4,003,901	4
Entity SameAs	142,448	5

Table 3. APIs and their descriptions.

API name	Description	Rank	Count
mention2entity	given mention name, return entity name	1	59,277,949
entityAVP	given entity name, return all its attribute-value pairs	2	35,812,420
entityTag	given entity name, return all its tags	3	27,334,278
entityInformation	given entity name, return its description information	4	22,698,972
entityType	given entity name, return all its types	5	17,608,266
entityAttribute	given entity and attribute names, return all values	6	36,936

We also publish a lot of APIs[2] to make our knowledge base accessible from Web. By December 2016, these APIs have already been called 164 million times since it is published on December 2015. Table 3 shows the function of each API and their usage statistics. We can see that `mention2entity` service which returns an entity for a certain mention of the entity in text is the most popular one. We envision that most of these APIs are serving many big data analytic applications as the underlying knowledge services.

6 Conclusion

In this paper, we propose a never-ending Chinese Knowledge extraction system: `CN-DBpedia`. Compared with other knowledge bases, `CN-DBpedia` relies quite few human efforts and provides the freshest knowledge with a smart active update strategy. The 160 million API calls of knowledge services provided by `CN-DBpedia` justify the rationality of our system design. We will further integrate more knowledge sources to increase the coverage of `CN-DBpedia` in the near future.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al. (eds.) ASWC/ISWC - 2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007). doi:10.1007/978-3-540-76298-0_52
2. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pp. 1247–1250. ACM (2008)
3. Cui, W., Xiao, Y., Wang, W.: KBQA: an online template based question answering system over freebase. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9–15 July, pp. 4240–4241 (2016)
4. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., et al.: Dbpedia-a large-scale, multilingual knowledge base extracted from wikipedia. Semant. Web J. **5**, 1–29 (2014)
5. Niu, X., Sun, X., Wang, H., Rong, S., Qi, G., Yu, Y.: Zhishi.me - weaving Chinese linking open data. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011. LNCS, vol. 7032, pp. 205–220. Springer, Heidelberg (2011). doi:10.1007/978-3-642-25093-4_14
6. Sabou, M., Bontcheva, K., Scharl, A.: Crowdsourcing research opportunities: lessons from natural language processing. In: Proceedings of the 12th International Conference on Knowledge Management and Knowledge Technologies, p. 17. ACM (2012)

[2] http://kw.fudan.edu.cn/cndbpedia/apiwiki/.

7. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web, pp. 697–706. ACM (2007)
8. Xie, C., Liang, J., Chen, L., Xiao, Y.: Towards End-to-End Knowledge Graph Construction via a Hybrid LSTM-RNN Framework
9. Xu, B., Zhang, Y., Liang, J., Xiao, Y., Hwang, S., Wang, W.: Cross-lingual type inference. In: Navathe, S.B., Wu, W., Shekhar, S., Du, X., Wang, X.S., Xiong, H. (eds.) DASFAA 2016. LNCS, vol. 9642, pp. 447–462. Springer, Cham (2016). doi:10.1007/978-3-319-32025-0_28
10. Yang, D., He, J., Qin, H., Xiao, Y., Wang, W.: A graph-based recommendation across heterogeneous domains. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 463–472. ACM (2015)