Building Knowledge Base through Deep Learning Relation Extraction and Wikidata

Pero Subasic, Hongfeng Yin and Xiao Lin

AI Agents Group, DOCOMO Innovations Inc, Palo Alto, CA, USA
{psubasic, hyin, xlin}@docomoinnovations.com

Abstract

Many AI agent tasks require domain specific knowledge graph (KG) that is compact and complete. We present a methodology to build domain specific KG by merging output from deep learning-based relation extraction from free text and existing knowledge database such as Wikidata. We first form a static KG by traversing knowledge database constrained by domain keywords. Very large high-quality training data set is then generated automatically by matching Common Crawl data with relation keywords extracted from knowledge database. We describe the training data generation process in detail and subsequent experiments with deep learning approaches to relation extraction. The resulting model is used to generate new triples from free text corpus and create a dynamic KG. The static and dynamic KGs are then merged into a new KB satisfying the requirement of specific knowledge-oriented AI tasks such as question answering, chatting, or intelligent retrieval. The proposed methodology can be easily transferred to other domains or languages.

Introduction

Knowledge graph (KG) plays an important role in closed domain question-answering (QA) systems. There are many large-scale KGs available (Bollacker 2008; Lehmann et al. 2012; Lenat 1995; Mitchell et al. 2018; Vrandecic and Krotzsh 2014). To answer user queries, a KG should be compact (pertain to a particular topic) or the QA engine may provide wrong answers due to the knowledge graph having too many extraneous facts and relations. The knowledge graph should be complete so as to have as many facts as possible about the topic of interest or the QA engine may be unable to answer user's query. The need for compactness and completeness are plainly at odds with each other such that existing KG generation techniques fail to satisfy both objectives properly. Accordingly, there is a need for an improved knowledge graph generation technique that satisfies the conflicting needs for completeness and compactness. We also aim to build a methodology to support easier knowledge base construction in multiple languages and domains.

We thus propose a methodology to build a domain specific KG. Figure 1 depicts the processes of domain specific KG generation through deep learning-based relation extraction and knowledge database. We choose Wikidata as the initial knowledge database. After being language filtered, the database is transformed and stored into MongoDB so that a hierarchical traversal starting from a set of seed keywords could be performed efficiently. This set of seed keywords can be given for specific application thus this approach can be applied to arbitrary domain. It is also possible to extract this set of keywords automatically from some given text corpora. The resulting subject-relation-object triples from this step are used to form a so-called static KG and also are used to match sentences from Common Crawl free text to create a large dataset to train our relation extraction model. The trained model is then applied to infer new triples from free text corpora which form a dynamic KG to satisfy the requirement of completeness. The static and dynamic KGs are then aggregated into a new KG that can be exported into various formats such as RDF, property graph etc., and be used by a domain specific knowledge-based AI agent.

The paper first reviews the related works regarding knowledge graph generation and relation extraction. It then describes our label dataset preparation, relation extraction model and KG generation in details, followed by some results of experiments of benchmarking relation extraction models and application of proposed approach for a soccer domain.

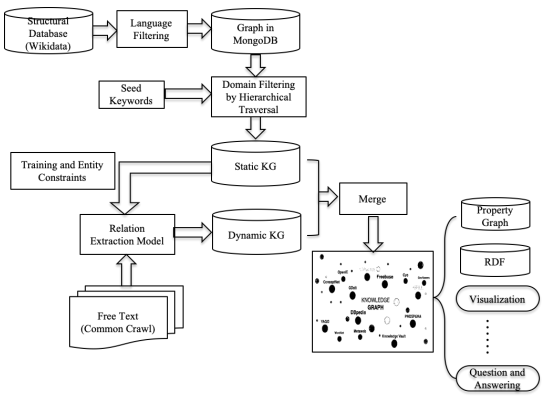


Figure 1. Flow Diagram for Construction of Domain Specific Knowledge Graph.

Related Work

A knowledge graph could be constructed by collaborative way to collect entities and links (Clark 2014), or automatic natural language processing to obtain subject-relation-object triples, such as through transformation of embedding representation (Lin et al. 2015; Socher et al. 2012; Wang et al. 2014), deep neural network model extraction approaches (Santos, Xiang and Zhou 2015; Zeng 2014; Zeng et al. 2015; Zhang and Wang; 2015; Zhou et al 2016) and inference method from graph paths (Guu, Miller and Liang et al. 2015). Researchers in recent years also propose to use end-to-end system (Kertkeidkachorn and Ichise 2017, Shang et al. 2019), deep reinforcement learning method (Feng 2018, Yang, Yang and Cohen 2017) to get better result.

As one of the major approaches to expand KG, relation extraction (RE) aims to extract relational facts from plain text between entities contained in text. Supervised learning approach is effective, but preparation of a high-quality labeled data is a major bottleneck in practice. One technique to avoid this difficulty is distant supervision (Mintz et al., 2009), which assumes that if two entities have a relationship in a known knowledge base, then all sentences that mention these two entities will express that relationship in some way. All sentences that contain these two entities are selected as training instances. The distant supervision is an effective method of automatically labeling training data. However, it has a major shortcoming. The distant supervision assumption is too strong and causes the wrong label problem. A sentence that mentions two entities does not necessarily express their relation in a knowledge base. It is possible that these two entities may simply appear in a sentence without the specific relation in the knowledge base. The noisy training data fundamentally limit the performances of any trained model (Luo et al. 2017). Most of RE researches focus on tiny improvements on the noisy training data. However, these RE results fall short from requirements of practical applications. The biggest challenge of RE is to automatically generate massive high-quality training data. We solve this problem by matching Common Crawl data with a structured knowledge base like Wikidata.

Our approach is thus unique in that it utilizes a structured database to form a static KG through hierarchical traversal of links connected with domain keywords for compactness. This KG is used to generate triples to train sequence tagging relation extraction model to infer new triples from free text corpus and generate a dynamic KG for completeness. The major contribution of our study is that we generated a large dataset for relation extraction model training. Furthermore, the approach is easily transferrable to other domains and languages as long as the text data is available. Specifically, to transfer to a new domain, we need a new set of keywords or documents representing the domain. To transfer to a new language, we need entity extractors, static knowledge graph in that language (Wikidata satisfies this requirement), and large text corpus in target language (Common Crawl satisfies that requirement, but other sources can be used).

Relation Extraction

Label Data Generation

The datasets used in distant supervision are usually developed by aligning a structural knowledge base like Freebase with free text like Wikipedia or news. One example is (Riedel, Yao, and McCallum 2010) who match Freebase relations with the New York Times (NYT) corpus. Usually, two entities with relation in a sentence associate a keyword in the sentence to represent the relation in the knowledge base. Therefore, it is required to match two entities and a keyword for a sentence to generate a positive relation. This will largely reduce noise in generating positive samples. However, the total number of positive samples is also largely reduced. The problem can be solved by using very large free text corpora: billions of web pages available in Common Crawl web data.

The Common Crawl corpus contains petabytes of data collected over 8 years of web crawling. The corpus contains raw web page data, metadata extracts and text extracts. We use one year of Common Crawl text data. After language filtering, cleaning and deduplication there are about 6 billion English web pages. The training data generation is shown in Fig. 2, and the in-house entity extraction system is used to label entities in Common Crawl text.

A Wikdata relation category has an id P-number, a relation name and several mapped relation keywords, for example:

- P-number: P19
- Name: place of birth
- Mapped relation keywords: birth city, birth location, birth place, birthplace, born at, born in, location born, location of birth, POB

Wikidata dump used in our task consists of:

- 48,756,678 triples
- 783 relation categories
- 2,384 relation keywords
-

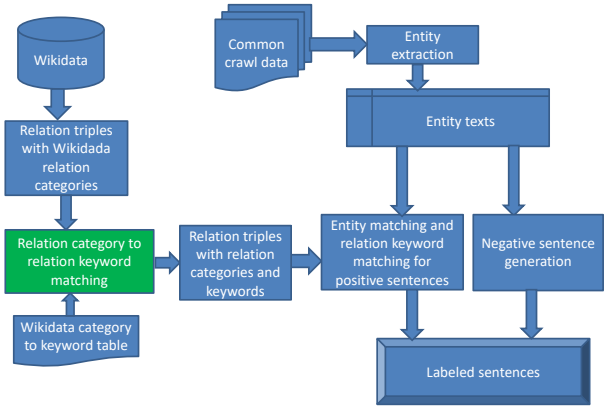


Figure 2. Flow Chart of Training Data Generation

First, Wikidata relation category triples are mapped to Wikidata relation keyword triples. Then, Wikidata keyword triples are matched with Common Crawl entity-labeled sentences. It yields:

- 386 million matched sentences
- 65 million unique sentences.
- There are 688 relation keywords with more than 1000 matched sentences
- Example:
  - Wikidata keyword triple:
    - [[Martín_Sastre]] born in [[Montevideo]]
  - Matched Common Crawl sentence:
    - [[Martín_Sastre]] was born in [[Montevideo]] in 1976 and lives in [[Madrid]]

Matched unique sentences for top relation keywords

- state 4,336,046
- city 4,251.983
- capital 2,797,477
- starring 2,032.749
- borders 1,874,461
- town 1,737,493
- wife 1,730,569
- founder 1,337,416
- is located in 1,136,473
- husband 1,016,505
- actor 1,014,708
- capital of 957,203
- son 954,848
- directed by 890,268
- married 843,009
- born in 796,941
- coach 736,866

Therefore, the massive high-quality labeled sentences are generated automatically for training supervised machine leaning models. With the labeled sentences, we can build RE models for specific domains, for specific relations or for open domain.

Relation Extraction Models for Soccer

In a specific domain example, we use labeled sentences to build RE models for soccer. First, we extract 17,950 soccer entities and 722,528 triples with at least one soccer entity from Wikidata, 78 relation categories with 640 relation keywords.

Training data generation:

- Positive sample generation:
  1. Select two entities (e1, e2) and a relation keyword (r_kw with relation category r_cat) in a matched sentence *s*
  2. If (e1, r_kw, e2) is in the relation keyword triples
  3. Set "e1, e2, r_kw, r_cat, *s*" as a positive sample
- Negative sample generation
  1. Select two entities (e1, e2) in a sentence s
  2. One entity must be a soccer entity
  3. Both entities are in the entity list generated from Wikidata relation triples
  4. Set "e1, e2, NONE, NA, *s*" as a negative sample. Select randomly with some probability to obtain sufficient number of negative samples.
  5. Remove duplicated samples
- Total Generated Training Data:
  - 2,121,640 samples
  - 335,734 positive relation sentences
  - 1,785,906 negative relation sentences

Building the Models

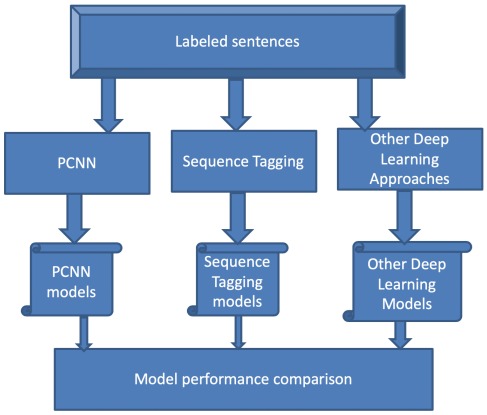


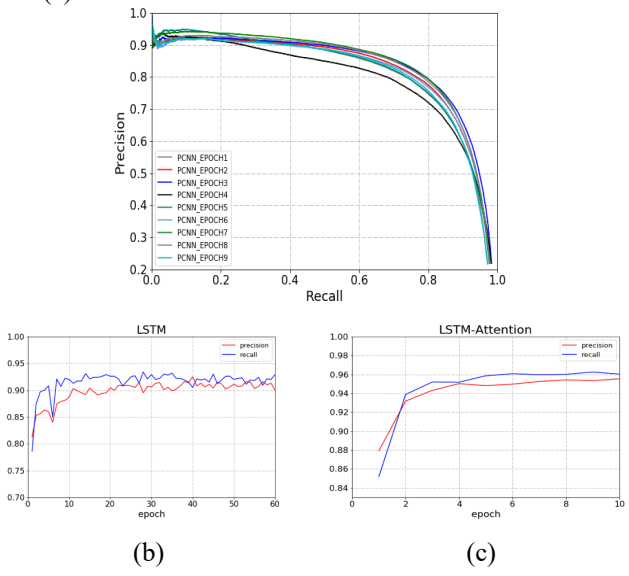
Figure 3. Flow Chart of Model Training Comparison

The PCNN model (Zeng et al. 2015), LSTM with Attention model (Zhou et al. 2016) and LSTM classification model (Zhang and Wang 2015) are trained with 90% data for training, 10% data for testing. Also, sequence tag model (Lample et al. 2016) is trained with 80% data for training, 10% data for testing during training and 10% data for testing after training.

A positive sentence is tagged as follows:

[[John]] Entity lives in [[New York]] Entity
O O O O B-Re I-Re O O O O O

(a)



(b) (c)

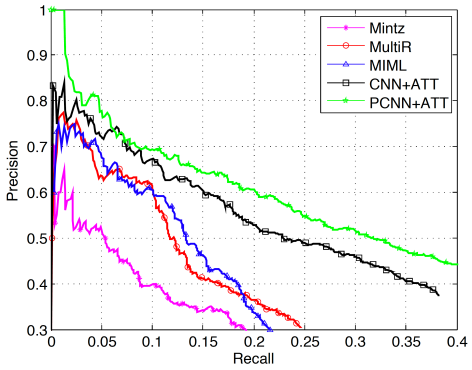
Figure 4. Performance Comparison (a) Precision vs Recall; (b) & (c) Precision vs Epoch

Table 1. shows the performance of each model. Based on F1 score: Sequence Tagging > LSTM+Attention > LSTM Classification > PCNN.

Model	F1	Precision	Recall
Sequence Tagging	**98.25%**	**97.61%**	**98.89%**
PCNN	82.89%	86.00%	80.00%
LSTM Classification	91.28%	90.10%	92.50%
LSTM + Attention	95.40%	94.80%	96.00%

Table 1. Performance Comparison of Different Models

In comparison with distant supervision datasets our datasets can train much higher-quality models.



Comparison of Figure 3 in Reference (Lin et al. 2016)

To validate the wining sequence tagging approach, we create validation data set from Common Crawl outside the training data by using different time period. We also validated on data used from other data source, different from Common Crawl with similar outcome. Validation results are as follows:

- F1 93.15%
- Precision: 90.43%
- Recall: 96.02%

Although the models perform well for the training data, we found that there are a lot of false positives when the models are applied on arbitrary free text. This issue can be improved with new negative sample generation which we describe here.

- Improved negative sample generation: a sentence *s* should have a keyword in the 640 relation keywords
  - For each pair of entities (e1, e2) in *s*
    - e1 or e2 is a soccer entity
    - e1 and e2 are in the entity list generated from Wikidata relation triples
    - If *s* is a matched sentence, and e1 and e2 are not in the relation triple of *s*
      - Set "e1, e2, NONE, NA, *s*" as a negative sample with a probability 0.5
    - If *s* is not a matched sentence

- Set "e1, e2, NONE, NA, *s*" as a negative sample with a probability 0.15 (15 out of 100 samples are selected as negative)
    - Remove duplicated samples
- The new training data with improved negative sample generation:
    - 1,702,924 samples
    - 363,458 positive samples
    - 1,339,466 Negative samples

With this training data, the performances of sequence tagging model on unseen data are reduced only slightly:

- F1: 92.38%
- Precision: 89.42%,
- Recall: 95.53%

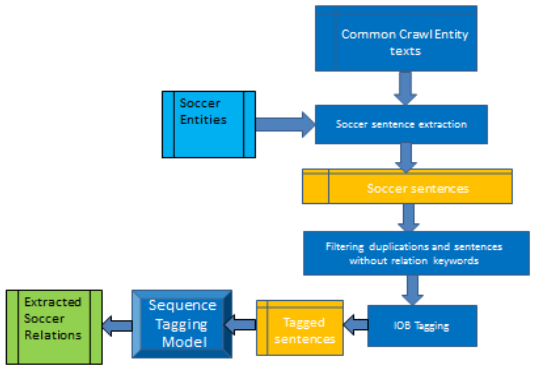


Figure 5. Soccer RE for Common Crawl data

Apply the model to Common Crawl data

Figure 5 shows the flowchart of soccer RE. For each sentence in Common Crawl entity texts, if the sentence contains at least one soccer entity and two entities in the entity list generated from Wikidata relation triples, the sentence is a soccer sentence. Then, the duplicated soccer sentences are removed and the sentences without the relation keywords are filtered out. The left sentences are tagged with IOB tags. Finally, the RE models are applied to the sentences to extract the relations. The results are:

- Total soccer sentences with two labeled entities: 64,085,913
- Total relations extracted: 600,964
- Aggregate unique relations: 147,486.

Construction of Knowledge Graph

As illustrated in Figure 1, the goal of the proposed approach is to build a knowledge graph from a static KG built from knowledge database and a dynamic KG generated from deep learning relation extraction. To form the static knowledge base, a suitable knowledge database (in this example, Wikidata) is language filtered (English, Japanese, and so on) and the resulting knowledge graph is stored in a suitable database platform such as MongoDB. To build the static knowledge graph, database is searched for seed keywords to act as seed vertices for the resulting knowledge graph. These seed vertices are then expanded by hierarchical traversal. In particular, the hierarchical traversal proceeds by finding all descendent vertices of the seed vertex. The algorithm then recursively iterates across these descendent (child) vertices. In addition, all ancestor vertices that have links to the seed vertex are identified by adding parent Wikidata items and recursively iterating across the parents of the seed vertex. Since the seed keywords are directed to the topic of interest (e.g., soccer), the hierarchical traversal of a resulting knowledge graph is also performing a domain filtering to the topic of interest. The relation triples from static knowledge graph may then be extracted and expanded to assist in the labeling of positive and negative sentences from a training corpus to train a deep learning relation extraction model. Deep learning model applied on free text, such as news articles, blogs, and similar up-to-date sources, generates dynamic knowledge graph. The static and dynamic knowledge graphs are then merged to form a combined knowledge graph. The two approaches ensure that we have slowly-changing (therefore 'static') knowledge as well as fast-changing (therefore 'dynamic') knowledge in the resulting knowledge graph. When merging static KG and dynamic KG, several subjective rules are enforced: (1) if relation of a triple in the dynamic KG is not defined in Wikidata property list, this triple will be ignored; (2) if any of two entities of a triple in the dynamic KG is not defined in the Wikidata item list, a pseudo item is created with a unique Q-number and the triple will be added into the knowledge base as a valid link; (3) relation defined in static KG has higher precedence – if a relation in dynamic KG is conflicted with a relation in static KG, the one in dynamic KG will be ignored and the relation in static KG will be kept in the merged knowledge base. The merged KG only exists in the final Neo4j database.

In our experiment, we assumed a single fact knowledge-based question answering system in soccer domain to demonstrate the proposed approach. To assure that the QA system can answer user query correctly, we make the KG contain facts represented by triples related with soccer as closely as possible. At the same time, we included as many

soccer related triples as possible. Wikidata is adopted as structured database to generate static KG. Relation extraction approach described in Section 2 is used to extract soccer related triples which are then merged into dynamic KG. Table 2 lists the top three relations in the new KG. It demonstrates that triple facts in the aggregated KG are correctly condensed into a specific domain of soccer. P641 (sport, in which the subject participates or belongs to) is not used in sentence labeling for its coverage is too broad. An example of triple with P641 is given: [Lionel Messi] => [P641 (sport)] => [association football].

P-number	Label	Occurrence
P641	sport	404,761
P54	member of sports team	128,632
P1344	Participant	30,307

Table 2. Top 3 Relations in Static Soccer KG

The statistics of static KG and dynamic KG is listed in Table 3. As it shows, static KG contains only 0.81% entities and 0.29% links from the original Wikidata. Queries performed on static KG thus will be significantly more efficient than on the original database, thus lowering requirement for computation power and memory usage. This is especially important for AI agent edge devices where hardware resources are limited. At the same time, the links in domain KG increased by 15.6%, resulting in a large increase of coverage. This number is dependent on the size of corpus used to extract relations. Larger corpus size will yield larger link increase resulting in more knowledge coverage. For example, in Wikidata database there are 67 links starting with Q170645 (2018 FIFA World Cup). In merged KG, this number increases to 472.

		Static KG	Merged KG
Number of Entities		405,639	425,224
Number of Predicates		676,500	807,718
% of Wikidata	Entity	0.81%	N/A
	Predicate	0.29%	
Increased Comparing to Static KG			15.6%

Table 3. Triple Statistics of Aggregated Knowledge Graph

Since there is no real question-answering system that is based on the knowledge graphs created in this study, improvement of question-answering performance from the merged KG over simply static KG or dynamic KG alone is not able to be evaluated quantitatively. Neo4j is used in the demonstration to simulate QA system – instead of a natural language question, a database query is issued to get response (in a real system this is usually accomplished by appropriate AIML mapping). Table 4 shows some query examples. As expected, some questions can be answered when merged KG is used because the corresponding facts are added from relation extraction results.

Q: who is Louis Giskus? A: [Louis Giskus] => [chairperson] => [Surinamese Football Association]
Q: how Antonio Conte is related with Juventus F.C.? A: [head coach]
Q: who is the manager of Manchester City F.C.? A: [Manchester City F.C.] => [represented by] => [Pep Guardiola]

Table 4. QA Examples using Dynamic KG

In Wikidata, defined items have different language labels. By incorporating corresponding language labels into Neo4j database, the resulting KB can easily accommodate the capabilities of visualizing or querying in languages other than English. As a demonstration, Figure 6 shows a query using Japanese to query the KB.

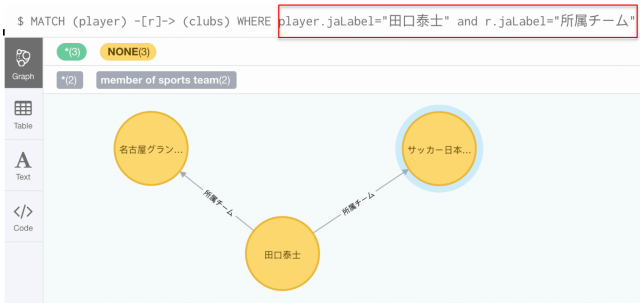


Figure 6. KB Query and Visualization in Japanese

Summary

This paper presents a methodology to build a knowledge graph for domain specific AI application where KG is required to be compact and complete. This KG is constructed by aggregating a static knowledge database such as Wikidata and a dynamic knowledge database, which is formed by subject-relation-object triples extracted from free text corpora through deep learning relation extraction model. In this study, a large high-quality dataset for training relation extraction model is developed by matching Common Crawl data with knowledge database. This dataset was used to train our own sequence tagging based relation extraction model and achieved the-state-of-art performance. Another important contribution is multi-language and multi-domain applicability of the approach.

It is inevitable that there might be wrong "facts" inferred from test corpora by the relation extraction model. It would be an interesting but challenging future work to evaluate validity of predicted triples and delete these wrong "facts" in order that they will not be integrated into knowledge base and become "truth". To infer new links directly from knowledge database to further expand the knowledge base could be another interesting topic. Another topic that could be worthy to pursue is to study whether joint named entity recognition and relation extraction could be integrated into our flow (Bekoulis et al. 2018).

Acknowledgments

We thank Yinrui Li for conducting the benchmark study of deep learning algorithms for relation extraction and contribution to the data of Figure 4. We also thank the anonymous reviewers for their helpful comments.

References

Bekoulis, G. et al. 2018. Joint Entity Recognition and Relation Extraction as a Multi-head Selection Problem. Expert System with Applications, vol 114, 34-45.

Bollacker, K. et al. 2008. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proceedings of SIGMOD'08*, 1247-1249, ACM

Clark, P. et al. 2014. Automatic Construction of Inference-Supporting Knowledge Bases. In *Proceedings of 4th Workshop on Automated Knowledge Base Construction* (AKBC'2014).

Feng, J. et al. 2018, Reinforcement Learning for Relation Classification from Noisy Data. in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, 5779-5786.

Guu., K., Miller, J. and Liang, P. 2015. Traversing Knowledge Graphs in Vector Space. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 318-327, Lisbon, Portugal.

Kertkeidkachorn, N. and Ichise, R. 2017. T2KG: An End-to-End System for Creating Knowledge Graph from Unstructured Text. In *Proceeding of AAAI-17 Workshop on Knowledge-Based Techniques for Problem Solving and Reasoning*, 743-749.

Lample, G. et al. 2016. Neural Architectures for Named Entity Recognition. In *Proceedings of NAACL-HLT 2016*, 260–270, San Diego, California.

Lehmann, J. et al. 2012. DBpedia – a Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web* 1(2012):1-5.

Lenat, D. 1995. CYC: A Large-Scale Investment in Knowledge Infrastructure. *Communications of the ACM* 38(11):33-38.

Mitchell, T. et al. 2018. Never Ending Learning. *Communications of the ACM* 61(5):103-115.

Lin, Y. et al. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, 2181-2187.

Lin, Y. et al. 2016. Neural Relation Extraction with Selective Attention over Instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2124-2133, Berlin, Germany.

Luo, B. et al. 2017. Learning with Noise: Enhance Distantly Supervised Relation Extraction with Dynamic Transition Matrix. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 430-439, Vancouver, Canada.

Mintz, M. 2009. Distant Supervision for Relation Extraction without Labeled Data. In *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, 1003-1011, Suntec, Singapore.

Riedel, S., Yao, L. and McCallum, A. 2010. Modeling Relations and Their Mentions without Labeled Text. In: Balcázar J.L., Bonchi F., Gionis A., Sebag M. (eds.) *Machine Learning and Knowledge Discovery in Databases*. *ECML PKDD 2010*. Lecture Notes in Computer Science, vol 6323. Springer, Berlin, Heidelberg.

Santos, C., Xiang, B. and Zhou, B. 2015. Classifying Relations by Ranking with Convolutional Neural Networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 626–634, Beijing, China.

Shang, C. et al., 2019. End-to-end Structure-Aware Convolutional Networks for Knowledge Base Completion, arXiv:1811.04441, accepted for Proceedings of AAAI 2019.

Socher, R. et al. 2012. Semantic Compositionality through Recursive Matrix-Vector Spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 1201-1211, Jeju Island, Korea.

Vrandecic, D. and Krotzsh M. 2014. Wikidata: A Free Collaborative Knowledgebase. *Communications of the ACM* 57(10):78-85.

Wang, Z. et al. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, 1112-1119.

Xie, Q. et al. 2017. An Interpretable Knowledge Transfer Model for Knowledge Base Completion. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 950–962, Vancouver, Canada, ACL.

Yang, F., Yang, Z. and Cohen, W. 2017. Differentiable Learning of Logical Rules for Knowledge Base Reasoning. In *Proceedings of 31st Conference on Neural Information Processing Systems* (NIPS 2017), Long Beach, CA, USA.

Zeng, D. 2014. Relation Classification via Convolutional Deep Neural Network. In *Proceedings of the 25th International Conference on Computational Linguistics* (COLING 2014), 2335–2344, Dublin, Ireland.

Zeng, D., Liu., K., Chen., Y. and Zhao, J. 2015. Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 1753–1762, Lisbon, Portugal, ACL.

Zhang, D and Wang, D. 2015. Relation Classification via Recurrent Neural Network. arXiv:1508.01006.

Zhou, P. et al. 2016. Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 207–212, Berlin, Germany, ACL.