

# Knowledge Extraction From Texts Based on Wikidata

Anastasia Shimorina

Johannes Heinecke

Frédéric Herledan

Orange, Lannion, France

firstname.lastname@orange.com

## Abstract

This paper presents an effort within our company of developing knowledge extraction pipeline for English, which can be further used for constructing an enterprise-specific knowledge base. We present a system consisting of entity detection and linking, coreference resolution, and relation extraction based on the Wikidata schema. We highlight existing challenges of knowledge extraction by evaluating the deployed pipeline on real-world data. We also make available a database, which can serve as a new resource for sentential relation extraction, and we underline the importance of having balanced data for training classification models<sup>1</sup>.

## 1 Introduction

Knowledge extraction aims at discovering semantic information from texts using a knowledge representation schema. This discovered information is used to build a knowledge base (KB), which is a useful resource for structured information. KBs can play an important role in many tasks and systems: domain question-answering systems, recommender systems, natural language generation systems, search result enhancement, and many others.

Enterprise knowledge bases have recently gained a lot of attention (Singhal, 2012; Liu et al., 2019a; Song et al., 2019; Dong et al., 2020). They allow to transform heterogeneous data, both public and private, into knowledge representations, which are effectively used for specific applications.

In this paper, we report on a preliminary step for building a company-specific KB, namely how to extract knowledge from texts in the form of (*subject, relation, object*) triples. We develop a system consisting of several components: entity detection and linking, coreference resolution, and relation extraction (RE). For the first two components we use

<sup>1</sup>The relation extraction database is available here: <https://github.com/Shimorina/relation-extraction-db-wikidata>

off-the-shelf tools, whereas for RE we develop our own module. Our RE module is based on Wikidata (Vrandečić and Krötzsch, 2014), the existing KB, which contains many pre-defined relations. With the goal to cover as many relations from Wikidata as possible, we create a database, which merges several datasets for RE and distributes them in a standardised format. We make use of this database to create different training scenarios for the RE task and show how balancing existing RE datasets impacts the task performance. We finally evaluate our knowledge extraction system on our company's internal data by human evaluation. Our system is deployed and is intended to be used on real-world data within the company.

Since we apply state-of-the-art NLP techniques, it is equally interesting to see the limitations of current approaches witnessed by our evaluation.

To summarise, the main contributions of this paper are the following:

- We provide insights based on real-world texts coming from industry, which allow to benchmark state-of-the-art systems on real-world data.
- We construct a database with cleaned and homogeneous datasets for sentence-based relation extraction from English texts.
- We train several models for the relation extraction task based on different training data and show how dataset balancing affects the task.
- We discuss some positive and negative results: what did and did not work in a real-life scenario.

## 2 Related Work

The literature on Information/Knowledge Extraction is incredibly vast (Martinez-Rodriguez et al., 2020). There exist many approaches for information extraction from raw texts. Here we describe

text	The abbreviation GDPR stands for “General Data Protection Regulation”. The GDPR governs the processing of personal data within the territory of the European Union. ...
triples	(GDPR <sub>Q1172506</sub> , instance of <sub>P31</sub> , abbreviation <sub>Q102786</sub> ) (General Data Protection Regulation <sub>Q1172506</sub> , short name <sub>P1813</sub> , GDPR <sub>Q1172506</sub> ) (GDPR <sub>Q1172506</sub> , main subject <sub>P921</sub> , personal data <sub>Q3702971</sub> ) (GDPR <sub>Q1172506</sub> , applies to jurisdiction <sub>P1001</sub> , European Union <sub>Q458</sub> )

Table 1: The beginning of text from the internal wiki page “GDPR”, and triples which correspond to the text. Wikidata IDs are given after subjects, objects, and relations. In the example all entities are linked to Wikidata, however it is not always possible.

different approaches for RE, the main subtask of information extraction.

We can differentiate between binary and  $n$ -ary relation extraction (Bach and Badaskar, 2007), which link two or more entities respectively. For triple extraction, most of research work concentrates on binary relation extraction (Sakor et al., 2020), however there are also approaches based on  $n$ -ary relation extraction, or semantic parsing, where different semantic formalisms are used. Frame Semantics, PropBank, Discourse Representation Structures, Abstract Meaning Representations were used to extract triples from texts (Gangemi et al., 2017; Fossati et al., 2018; Mihindukulasooriya et al., 2020).

Another important difference in RE approaches is the use of an open or closed relation set. A closed, pre-defined set of relations is targeted in relation classification systems, where either a custom pool of relations is used (Gábor et al., 2018) or the set is defined by an underlying KB, such as Wikidata, Freebase, DBpedia, etc. The paradigm opposite to closed RE is Open Information Extraction (Etzioni et al., 2008, OpenIE). OpenIE aims at extracting domain-independent relations from large corpora without using a predefined schema. OpenIE systems may extract redundant information due to lexical variations in texts, so while using this paradigm for knowledge extraction, a process called canonicalisation is used to reconcile their output with a given KB (Lin et al., 2020).

We adopt the knowledge extraction approach with binary RE on a closed set of relations from Wikidata. We hope that this choice will allow us to facilitate enterprise-specific KB construction in the future.

### 3 Data

Our company has an internal wiki in English where different terms are explained. Those terms can

belong to some general knowledge (e.g., *climate change*, *Agile software development*) or can be specific to the company. The wiki terms span over several areas: from human resources, marketing, legal affairs to computer science and information technology. Those wiki documents, while having valuable information for the company, represent unstructured text with no linguistic annotation; they sometimes exhibit some information overlap or they can have related term descriptions span over several pages not linked between each other. An example of the beginning of such document describing a general term is displayed in Table 1. The length of a document is variable: it can range from several to a few hundred sentences.

Our motivation to explore internal documents is as follows: we would like to represent the information in a structured way, that would allow reasoning and better understanding of the company knowledge. Moreover, a potential KB may serve in different downstream applications developed in the company: question answering, task-oriented dialogue, knowledge management.

We chose Wikidata as our initial KB schema because of its steady growth within last years and increased community participation. The Wikidata schema may be eventually refined to better suit our needs in the future. For instance, we might add new relations, not yet defined in Wikidata.

### 4 Approach Overview

We aim to extract RDF (Resource Description Framework) triples in the form (*subject*, *relation*, *object*) from text.

We develop a classical pipeline for triple extraction: sentence splitting, entity detection and linking, coreference resolution, and relation classification.

1. Text is preprocessed and is split into sentences with pySBD (Sadvilkar and Neumann, 2020).

dataset	# instances	# R types	% neg.	human checks	license
FewRel (Han et al., 2018)	56,000	80	0%	yes	MIT
T-REx (Elsahar et al., 2018)	12,081,023	652	0%	no	CC BY-SA 4.0
DocRED (Yao et al., 2019)	778,914	96	0%	no (yes*)	MIT
WikiFact (Goodrich et al., 2019)	33,628,338	934	92%	no	CC BY 4.0
Wiki20m (Han et al., 2020)	738,463	81	60%	no	MIT
WebRED (Ormandi et al., 2021)	107,819	385	54%	yes	CC BY 4.0
our database (DB)	47,390,557	1,022	66%	yes/no	CC BY-SA 4.0

Table 2: Summary of the datasets used in the database. R types is a number of relation types including P0 and NA; neg. is a percentage of negative examples, i.e. examples with no relation detected or unknown relation; human checks correspond to whether some human checks were carried out to construct a dataset. \*The large part of DocRED is collected using distant supervision; 2.68% of the dataset instances were verified by humans.

- Entities are detected and linked to Wikidata IDs with GENRE (Cao et al., 2021). We chose GENRE because it identifies common nouns as well as proper nouns and links them to Wikidata. Common noun identification was important for our case, since the texts under consideration often describe common noun terms rather than named entities, such as geographical locations, persons, which frequently are the target of other popular named entity recognisers. GENRE is also able to identify entities without linking them to Wikidata. That feature was useful for us while handling texts about company-specific named entities and abbreviations.
- Coreference is resolved with neuralcoref from HuggingFace<sup>2</sup>.
- For each pair of entities ( $e_1$ ,  $e_2$ ) present in a sentence, a Wikidata relation is predicted using our relation classifier. It predicts whether a relation exists, and if yes, which one.

The desired output of the pipeline is shown in Table 1.

While for the steps 1-3, we used off-the-shelf libraries, for relation extraction we developed an in-house solution, which is described in Section 5.

## 5 Relation Extraction

RE is a notoriously difficult task because relations, as compared to entities, are not often expressed explicitly, i.e. it is hard to find a precise verbal expression. Moreover, relations can be expressed in many different ways in a text. RE usually works

well when it covers a limited set of well-defined relations. We, on the other hand, have no explicit relations, defined on our data. Our goal is to explore texts to possibly find some relations coming from the external closed set (Wikidata).

Relations in Wikidata are numerous. There are around 9,500 relations as of January 2022<sup>3</sup>. However, most of them are not exploitable in ordinary texts, since a lot of them are about some ID numbers in different catalogues and libraries, e.g. IMDb ID (P345), Swiss parliament ID (P1307), etc. We estimate that about 1,500-2,000 relations can be usable in everyday texts. To explore relations in Wikidata, we use available datasets for RE that are based on Wikidata relations. It means that we could not use other popular datasets for RE such as NYT (Riedel et al., 2010) or TACRED (Zhang et al., 2017), since they use other knowledge bases.

### 5.1 Database

Within the RE task modelling, our goal was to have as many relations from Wikidata as possible to increase the probability to find relations in our data. The issue with most datasets for RE is that relation types are few. So we proceeded to create a database (DB)—a common resource where several RE datasets are merged.

We preprocessed 6 existing datasets to adapt them to sentence-based RE (see Table 2): FewRel (Han et al., 2018), T-REx (Elsahar et al., 2018), DocRED (Yao et al., 2019), WikiFact (Goodrich et al., 2019), Wiki20m (Han et al., 2020), and WebRED (Ormandi et al., 2021). Initially these datasets were developed for different purposes and with different methods. Most of them were collected using

<sup>2</sup><https://github.com/huggingface/neuralcoref>

<sup>3</sup><https://www.wikidata.org/w/index.php?title=Special:ListProperties>

instance	relation	label
SUBJ{Under Pressure} is a 1981 song by Queen and OBJ{David Bowie}.	P676	lyrics by
Official figures showed there were 25 million baptised Anglicans in OBJ{England} and SUBJ{Wales}.	P0	no relation

Table 3: Examples of database instances where the subject and object are marked with special symbols in the text. Labels correspond to Wikidata relation labels.

distant supervision (Mintz et al., 2009), afterwards some of them were verified by humans. After merging those datasets, we obtain a dataset with 1,022 unique Wikidata relation types including the relation ‘P0’ (called negative relation), which means “the absence of relation” between the designated subject and object, and ‘NA’, which defines an unknown relation. The main advantage of the created DB is to have a homogeneous dataset where an instance is a sentence with a subject and an object identified and a relation between them (see examples in Table 3). Apart from this main information, the DB stores some additional features that were available in the original datasets, e.g. Wikidata IDs for subjects and objects, a source document for the sentence, etc. If training/validation/test split was provided for a dataset, we did not include test splits in the DB to reduce possible overuse of test data by future users. We also ensure that datasets coming from the same research groups do not have an overlap by deleting duplicate items.

The DB is easy and fast to query to obtain a sample of desired data: for example, choosing the instances that were verified by humans, choosing the instances expressing a particular relation, etc. We hope that the DB will serve the community by providing an easy access to RE datasets standardised for the sentence-based RE task. We will make it available upon acceptance.

## 5.2 Training Data

While most RE datasets were collected automatically, WebRED presents a cleaned dataset with the most relation coverage, so we use it as a main source for training and testing our RE models. We also know that around 50% of examples are negative examples in WebRED, i.e. examples with ‘P0’, so we paid special attention to that while constructing our training data. We used four collections of training data to develop different models for the RE component of our pipeline:

1. **WebRED**. It is the original dataset, called WebRED<sub>H2+1</sub> in Ormandi et al. (2021). The training data contains 383 relations (classes)

for relation classification; there are 2 classes less than shown in Table 2 because they happened to appear only in the validation part.

2. **WebRED-balanced**. WebRED, as it is also often the case with other RE datasets, is largely imbalanced: 30 most frequent relation types cover more than 90% instances in the dataset. So for each relation that has less than 500 examples we tried to add more examples from other corpora present in the DB to reach 500 examples per relation if possible. This training data has 385 classes. After adding the underrepresented relations, 30 most common relation types account for 43% of instances in the dataset.
3. **DB-500**. In this case we aim to explore all the relation types present in the DB. For each relation (including P0 but excluding NA), we choose 500 training examples from different datasets, preferably choosing in the first place from the datasets where human annotation was present. However, a relation can still have less than 500 examples for training if there are not enough examples in the DB. This training data has 1,013 classes. From 1,022 relations present in the DB (Table 2), we removed NA and 8 Wikidata relations that existed in Wikidata during the time of dataset creation but that were subsequently removed from Wikidata.
4. **DB-500+neg**. As we test our approach on WebRED development data where negative examples constitute more than a half of the dataset, we add all the P0 examples from WebRED to DB-500. This training set equally has 1,013 classes.

Number of training instances for each training set is shown in Table 4. In what follows, we do not compare our results to the numbers reported in Ormandi et al. (2021), since the published dataset is different from the one used in their paper due to

training data	# examples	# classes	F1	P	R	F1*	P*	R*
WebRED	97,037	383	80.47	80.48	80.47	72.87	71.79	73.99
WebRED-balanced	215,937	385	85.65	85.90	85.39	80.02	77.86	82.30
DB-500	205,331	1,013	49.60	51.81	47.58	51.17	41.84	65.85
DB-500+neg	249,532	1,013	69.14	69.88	68.42	53.70	65.06	45.71

Table 4: Classification results on WebRED validation data. Classes include the P0 relation. F1: micro F1; P: precision; R: recall. \* negative examples were removed from the evaluation data. When training with 5 random seeds, the standard deviation in the range of 0.14-0.59 was observed for the scores.

copyright<sup>4</sup>.

### 5.3 Experimental Setup

**Computational experiments.** We treated RE as a multi-class classification problem where one relation must be predicted given a set of all possible relations. We fine-tune RoBERTa<sub>LARGE</sub> (Liu et al., 2019b) by adding a softmax classification layer, and we use the training data described in Section 5.2. Each training instance has special symbols around subject and object entities (see examples in Table 3). We use the simpletransformers library<sup>5</sup>, which in its turn is built on the HuggingFace Transformers library (Wolf et al., 2020). The models are fine-tuned with the AdamW optimiser (Loshchilov and Hutter, 2019), with a learning rate of 0.00004, and a batch size of 32 for three epochs. Models were trained on two GPUs (GeForce GTX 1080 Ti); training time ranged from three to seven hours depending on the size of training data.

Evaluation was done on WebRED development data, which has 10,782 instances. We did not use WebRED test data, since we think of continuing our model development.

**Human evaluation.** We assessed the performance on our unlabeled data (see Section 3) with one human annotator. We focused on entity detection and linking, and RE. For entity recognition, one of the authors of the paper examined 52 first paragraphs of the wiki documents, where 550 entities and 457 linked entities were tagged by GENRE. For RE, the annotator examined 100 relations predicted by the model trained on DB-500+neg with the highest probability scores (more than 0.94). The relations were assessed on a 3-point scale (1: “bad”, 2: “not sure; ambiguous case”, 3: “good”).

<sup>4</sup>See Ormandi et al. (2021) and <https://github.com/google-research-datasets/WebRED>

<sup>5</sup><https://github.com/ThilinaRajapakse/simpletransformers>

## 6 Results

### 6.1 Entity Detection and Linking

	F1	P	R
Entity Detection	0.78	0.83	0.74
Entity Linking	0.71	0.79	0.64

Table 5: Manual evaluation. Micro F1, precision, recall for entity detection and linking on our data.

Manual evaluation of entity detection and linking based on GENRE without any fine-tuning on our data showed quite satisfying performance (Table 5). Entity detection reaches F1 of 0.78 with high precision of 0.83. Entity linking performs a bit worse with F1 of 0.71 and precision of 0.79. We conjecture that this relatively good performance may be due to the resemblance of our data (factual documents) to Wikipedia texts, which were used for training of the entity recogniser.

### 6.2 RE on WebRED Validation Data

Table 4 presents the results of the classification task. Models fine-tuned with WebRED show higher scores due to the lower number of classes. The highest micro F1 (85.65) is achieved with the balanced version of WebRED. Overall, all the metrics are higher for WebRED-balanced, which suggests that simply adding more examples for underrepresented classes could help notably increase overall performance. Models learned on DB have more classes to predict, hence lower scores comparing to WebRED-based fine-tuning. F1 drastically drops to 49.60 in the case of DB-500. DB-500+neg shows better results for all metrics as compared to DB-500; this finding highlights the importance of accounting for the majority negative class present in the evaluation data.

The performance without the negative examples, which represent more than 50% of the evaluation

sentence	relation	label	human ann.
Venture capitalists refers to specialized SUBJ{professional} OBJ{investors} who generally invest the money of institutional investors in early-stage startups.	P425	field of this occupation	bad
The Kaya equation was developed by SUBJ{Yoichi Kaya}, a Japanese energy economist, in his book OBJ{"Environment, Energy, and Economy: Strategies for Sustainability."}.	P800	notable work	good
Civil liberties refer to all OBJ{individual} and SUBJ{collective} rights and freedoms guaranteed by the State and regulated and protected by law.	P461	opposite of	not sure

Table 6: Human annotation of model predictions for the RE task.

data, is also shown (marked with \*). We can see that in the case of WebRED the performance drops from 80.47 to 72.87 as measured by F1; the drop in WebRED-balanced is twice less — 5 points — due to the more balanced nature of the training data. Naturally, DB-500, being the dataset without a negative majority class, does not yield any drop: on the contrary, F1 increases from 49.60 to 51.17. DB-500+neg where a large set of negative examples were added exhibits the opposite trend: F1 goes down from 69.14 to 53.70.

Precision was a little bit higher than recall in the negative example evaluation setting (P vs. R); however, without the negative examples (P\* vs. R\*), recall was higher for WebRED, WebRED-balanced, and DB-500. In the case of DB-500+neg, precision is 65.06 while recall is lower (45.71).

### 6.3 RE on Real-World Data

Despite evaluating the relations with high probability scores, the human evaluation results are less confident: 70% of examples were tagged as “bad”, 19% as “good”, and 11% as “not sure; ambiguous case”. Some examples of model prediction along with human ratings are shown in Table 6. Most of the examples annotated as “bad” are connected to entity detection, which was not pertinent for RE, as in the case of the first example in Table 6. Annotating an example as “not sure” usually means that the relation is not explicitly conveyed in a sentence, and it is hard to say whether it is present or not (see the third example). Overall, we witness that RE is a much more difficult task than entity detection and linking for existing methods when they are applied to the data that was never seen during training.

### 6.4 Final Pipeline and Time-Task Distribution

The described pipeline of knowledge extraction from text is deployed within our company and can be executed on any corpus of texts. It is hosted on a virtual machine with 4 CPU and 32 Gb of RAM, and all the pipeline components are ran on

CPU. The time of response is not immediate, but in our use case we do not consider it important. To develop the pipeline, we spent most of the time working with data rather than developing models. Here is the approximate time-task distribution:

- 3 weeks: overall design (understanding the task and the needs, related work review, pipeline conception, looking at our data)
- 1 week: entity detection/linking and coreference
- 5 weeks: RE data preparation (search for corpora, collect, clean, prepare)
- 3 weeks: RE model development
- 1 week: entity detection/linking and RE evaluation
- 3 weeks: final pipeline deployment (code cleaning/refactoring, dockerisation, API, documentation)

## 7 Conclusion

In this paper we presented the pipeline for knowledge extraction from text based on Wikidata. We showed its utility on real-world data coming from our company’s internal wiki. While entity detection and linking tools perform well on unseen data, RE still presents significant challenges. We developed the database for sentence-based RE with a large coverage of Wikidata relations, which we hope will be useful for the community. We also showed that balancing training data is crucial for good performance in RE.

In future work, we plan to improve the current pipeline, especially the RE component. We envisage several possible ways that can be explored: integrating syntactic parsing for better entity detection, using insights from semantic parsing to better represent sentence structure, and annotating some part of our data and using it for fine-tuning.

## References

- Nguyen Bach and Sameer Badaskar. 2007. A review of relation extraction. *Literature review for Language and Statistics II*, 2:1–15.
- Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. [Autoregressive entity retrieval](#). In *International Conference on Learning Representations*.
- Xin Luna Dong, Xiang He, Andrey Kan, Xian Li, Yan Liang, Jun Ma, Yifan Ethan Xu, Chenwei Zhang, Tong Zhao, Gabriel Blanco Saldana, Saurabh Deshpande, Alexandre Michetti Manduca, Jay Ren, Suresnder Pal Singh, Fan Xiao, Haw-Shiuan Chang, Giannis Karamanolakis, Yuning Mao, Yaqing Wang, Christos Faloutsos, Andrew McCallum, and Jiawei Han. 2020. [AutoKnow: Self-Driving Knowledge Collection for Products of Thousands of Types](#), page 2724–2734. Association for Computing Machinery, New York, NY, USA.
- Hady Elsahar, Pavlos Vougiouklis, Arslan Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. [T-REx: A large scale alignment of natural language with knowledge base triples](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. 2008. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74.
- Marco Fossati, Emilio Dorigatti, and Claudio Giuliano. 2018. N-ary relation extraction for simultaneous t-box and a-box knowledge base augmentation. *Semantic Web*, 9(4):413–439.
- Kata Gábor, Davide Buscaldi, Anne-Kathrin Schumann, Behrang QasemiZadeh, Haifa Zargayouna, and Thierry Charnois. 2018. [SemEval-2018 task 7: Semantic relation extraction and classification in scientific papers](#). In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 679–688, New Orleans, Louisiana. Association for Computational Linguistics.
- Aldo Gangemi, Valentina Presutti, Diego Reforgiato Recupero, Andrea Giovanni Nuzzolese, Francesco Draicchio, and Misael Mongiovi. 2017. Semantic web machine reading with fred. *Semantic Web*, 8(6):873–893.
- Ben Goodrich, Vinay Rao, Peter J. Liu, and Mohammad Saleh. 2019. [Assessing the factual accuracy of generated text](#). In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD ’19, page 166–175, New York, NY, USA. Association for Computing Machinery.
- Xu Han, Tianyu Gao, Yankai Lin, Hao Peng, Yaoliang Yang, Chaojun Xiao, Zhiyuan Liu, Peng Li, Jie Zhou, and Maosong Sun. 2020. [More data, more relations, more context and more openness: A review and outlook for relation extraction](#). In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 745–758, Suzhou, China. Association for Computational Linguistics.
- Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. [FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.
- Xueling Lin, Haoyang Li, Hao Xin, Zijian Li, and Lei Chen. 2020. [Kbpearl: A knowledge base population system supported by joint entity and relation linking](#). *Proc. VLDB Endow.*, 13(7):1035–1049.
- Danyang Liu, Ting Bai, Jianxun Lian, Xin Zhao, Guangzhong Sun, Ji-Rong Wen, and Xing Xie. 2019a. [News graph: An enhanced knowledge graph for news recommendation](#). In *Proceedings of the Second Workshop on Knowledge-aware and Conversational Recommender Systems, co-located with 28th ACM International Conference on Information and Knowledge Management, KaRS@CIKM 2019, Beijing, China, November 7, 2019*, volume 2601 of *CEUR Workshop Proceedings*, pages 1–7. CEUR-WS.org.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. [Roberta: A robustly optimized bert pretraining approach](#).
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.
- Jose L Martinez-Rodriguez, Aidan Hogan, and Ivan Lopez-Arevalo. 2020. Information extraction meets the semantic web: a survey. *Semantic Web*, 11(2):255–335.
- Nandana Mihindukulasooriya, Gaetano Rossiello, Pavan Kapanipathi, Ibrahim Abdelaziz, Srinivas Ravishankar, Mo Yu, Alfio Gliozzo, Salim Roukos, and Alexander Gray. 2020. [Leveraging semantic parsing for relation linking over knowledge bases](#). In *The Semantic Web – ISWC 2020*, pages 402–419, Cham. Springer International Publishing.
- Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. [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*, pages

- 1003–1011, Suntec, Singapore. Association for Computational Linguistics.
- Robert Ormandi, Mohammad Saleh, Erin Winter, and Vinay Rao. 2021. [Webred: Effective pretraining and finetuning for relation extraction on the web](#).
- Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases: Part III, ECML PKDD'10*, page 148–163, Berlin, Heidelberg. Springer-Verlag.
- Nipun Sadvilkar and Mark Neumann. 2020. [PySBD: Pragmatic sentence boundary disambiguation](#). In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, pages 110–114, Online. Association for Computational Linguistics.
- Ahmad Sakor, Kuldeep Singh, Anery Patel, and Maria-Esther Vidal. 2020. [Falcon 2.0: An Entity and Relation Linking Tool over Wikidata](#), page 3141–3148. Association for Computing Machinery, New York, NY, USA.
- Amit Singhal. 2012. [Introducing the knowledge graph: things, not strings](#).
- Dezhao Song, Frank Schilder, Shai Hertz, Giuseppe Saltini, Charese Smiley, Phani Nivarthi, Oren Hazai, Dudi Landau, Mike Zaharkin, Tom Zielund, Hugo Molina-Salgado, Chris Brew, and Dan Bennett. 2019. [Building and querying an enterprise knowledge graph](#). *IEEE Transactions on Services Computing*, 12(3):356–369.
- Denny Vrandečić and Markus Krötzsch. 2014. [Wikidata: A free collaborative knowledgebase](#). *Commun. ACM*, 57(10):78–85.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. [DocRED: A large-scale document-level relation extraction dataset](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, Florence, Italy. Association for Computational Linguistics.
- Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. [Position-aware attention and supervised data improve slot filling](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.