A Semantics-aware Transformer Model of Relation Linking for Knowledge Base Question Answering

Tahira Naseem, Srinivas Ravishankar, Nandana Mihindukulasooriya, Ibrahim Abdelaziz, Young-Suk Lee, Pavan Kapanipathi, Salim Roukos, Alfio Gliozzo, Alexander Gray
IBM Research
{tnaseem, ysuklee, kapanipa, roukos, gliozzo}@us.ibm.com
{srini, nandana.m, ibrahim.abdelaziz1, alexander.gray}@ibm.com

Abstract

Relation linking is a crucial component of Knowledge Base Question Answering systems. Existing systems use a wide variety of heuristics, or ensembles of multiple systems, heavily relying on the surface question text. However, the explicit semantic parse of the question is a rich source of relation information that is not taken advantage of. We propose a simple transformer-based neural model for relation linking that leverages the AMR semantic parse of a sentence. Our system significantly outperforms the state-of-the-art on 4 popular benchmark datasets. These are based on either DBpedia or Wikidata, demonstrating that our approach is effective across KGs.

1 Introduction

Knowledge base question answering (KBQA) has received significant interest due to its real-world applications. KBQA is a task where a natural language question is transformed into a precise structured query, using Entity Linking and Relation Linking as necessary sub-tasks to retrieve an answer. For example, the question *"Who founded the city where Pat Vincent died?"* requires mapping (a) *founded* and *died* to relations *dbo:founder* and *dbo:deathPlace*, and (b) entity *Pat Vincent* to *dbr:Pat_Vincent*, given DBpedia as the knowledge base.

Semantic parses such as Abstract Meaning Representation (AMR) have recently shown to be useful for the KBQA task (Lim et al., 2020). However, critical tasks for KBQA such as Relation Linking continue to be addressed primarily using the question text (Mulang' et al., 2020; Sakor et al., 2019b; Lin et al., 2020), ignoring the AMR parses of the question which can introduce additional semantics. In the literature, some systems such as SLING (Mihindukulasooriya et al., 2020) have used AMR for relation linking. However, similar to other rule-based approaches (Sakor et al., 2019b), SLING depends heavily on the specific target KG (DBpedia) and it is based on a complex ensemble of different approaches, making portability to new knowledge bases a non-trivial task.

In this work, we propose SemReL; a single Semantics-aware neural model for Relation linking. SemReL takes as input the question text annotated with its AMR parse and entity information and outputs a ranked list of relations. The key contributions of this work are as follows: (a) a simple, knowledge graph agnostic neural model for relation linking over knowledge bases, (b) leveraging AMR parses for better question representation, and (c) an experimental evaluation using four datasets based on DBpedia and Wikidata where we show that SemReL consistently outperforms existing systems on all datasets.

2 Semantics-aware Relation Linking

We propose a relation linking system that exploits the semantic structure of a sentence to retrieve relevant relations from the underlying knowledge base. We hypothesise that semantic representations abstract away from lexical forms, providing structural clues that are more consistent across training examples than surface text. To this end, we use the AMR graph of the sentence as its semantic structure. AMRs are directed acyclic graphs that capture *who is doing what to whom* in a sentence. The nodes in the graph are concepts and the edges are labelled with relations between those concepts. Figure 1 shows example AMR graphs for the question *"Who founded the city where Pat Vincent died?"*. Note that the AMR graph for the question represents the target of the query as a special node labelled 'amr-unknown'.

The inputs to our system are: the question text,

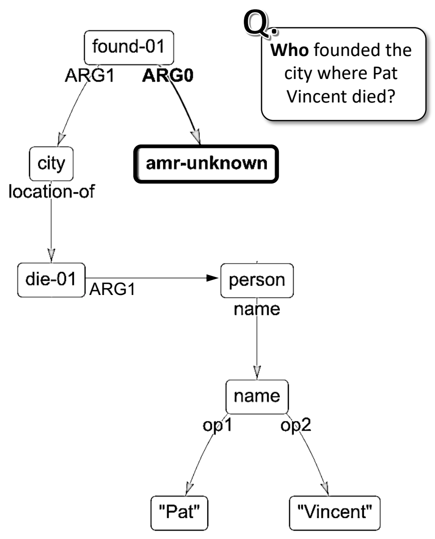

Figure 1: A question AMR: The *amr-unknown* represents the query target and the *name* node marks entities.

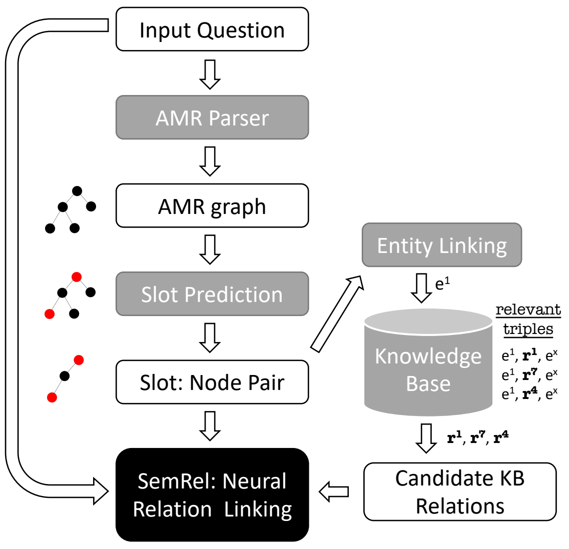

Figure 2: Overall system flow: grey blocks are various systems and white blocks show the inputs and outputs.

its AMR graph and the entities in the question marked and linked[1]. Relation linking is performed in two steps. First, our system identifies the number of expected relations and their location both in the sentence and in the AMR graph. Next, for each identified slot, the most likely relation is predicted using a transformer based neural model, that ranks them using their English labels from the KG. The AMR structure of the sentence is crucial in both steps. Figure 2 shows a schematic diagram of overall system. In the following, we first explain the process of finding potential relation slots using AMR graph. Next we describe in detail our relation linking module.

2.1 Relation Slot Prediction

AMR explicitly marks named entity nodes (see Figure 1). These nodes are linked to knowledge base entities using BLINK entity linker. The entity nodes in graph are also used to predict the number and locations of relation slots. A *slot* is defined as a pair of nodes in the AMR graph, where the corresponding entities have a relation in knowledge base in the context of the question. For instance, in Figure 3, nodes *city* and *person* are involved in a KB relation *death place* relevant for this question. Slot prediction is done using a deterministic rule-based transformation described in (Kapanipathi et al., 2021). In particular, we use their path-based approach where all the paths between the *amr-unknown* node and the linked entity nodes are retrieved. Then all node pairs along the path that are joined by a predicate node are considered a relation slot. We refer the reader to the original publication for more details of the method.

2.2 Neural Relation Linking Model

SemReL employs a Siamese network, where the input question and target relations are embedded in the same vector space. The most likely relation is the one whose representation is closest to that of the input question. Figure 3 shows the overall architecture of our model. We use a Transformer model (Vaswani et al., 2017) as a shared encoder for both the input questions and candidate relations. In particular, we use the pre-trained BERT model (Devlin et al., 2018) to initialize the encoder parameters. The output vector corresponding to the starting [CLS] token is used as the vector representation of the input. This vector is passed through a feed-forward linear layer that projects it to the shared embedding space. Unlike the transformer parameters, the weights of the linear projection layer on top are *not* shared between the questions and the relations.

Semantic information is given as part of the question input to the encoder. As mentioned above, during the preprocessing step, the pairs of nodes in AMR graph are identified for relation linking. For instance, in figure 3, the nodes ‘person’ and ‘city’ are marked in the input graph as the participants

[1]We use the stack transformer parser of Astudillo et al. (2020); Lee et al. (2020) for generating AMR graphs and the BLINK system of Wu et al. (2019) for entity linking.

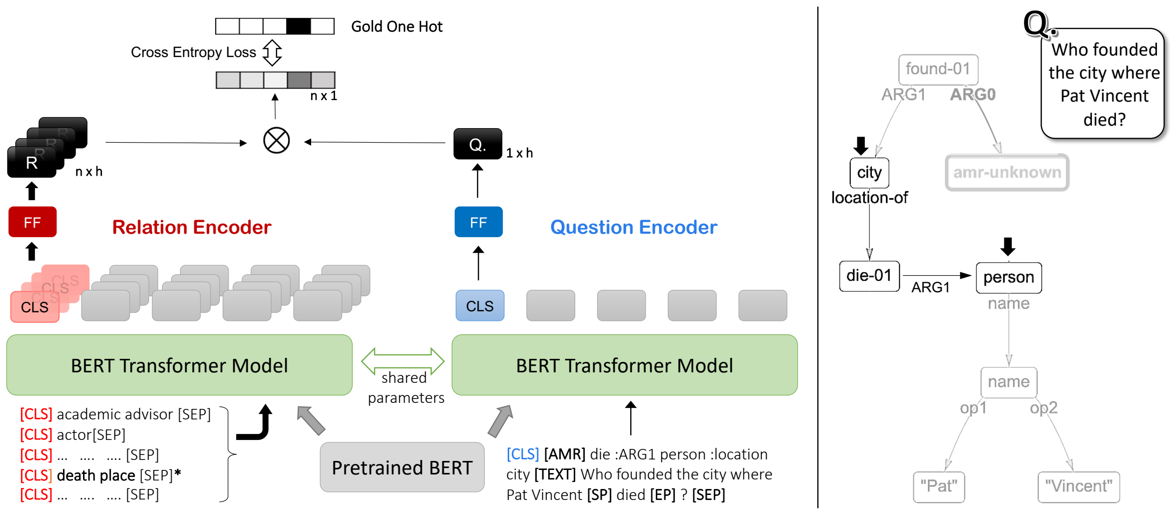

Figure 3: SemReL model architecture (left) and inputs to the model (right).

of a potential relation. The subgraph connecting these nodes is traversed in a top-down manner to form a linearized representation; in this case, it will yield the linearized string '*die :ARG1 person :location city*'. Note that the sense label of the node 'die-01' is dropped. Moreover, all reversed AMR relations with *-of* suffix are normalized to their original relation name and direction. In this example *:location-of* is mapped to *:location* with direction reversed. We prepend this linearized AMR path string to the input question text along with a special leading token [AMR]. The question text also starts with a special leading token [TEXT]. The word aligned to the root of the AMR subgraph is marked as the predicate[2], using special start and end predicate tokens [SP] and [EP].

Figure 3 shows the complete input for the example question that goes into the Question Encoder. The same transformer model also serves as relation encoder. Relation names are tokenized using BERT tokenizer without any additional pre-processing. We add special tokens [AMR], [TEXT], [SP] and [EP] as well as the AMR relation labels into the BERT vocabulary.

Training Objective: During training, for each example, scores are computed for the gold relation as well as a set of negative examples based on the inner product of their vectors with that of the question. For a relation r_i with vector representation $\mathbf{r_i}$ and a question q_n with vector representation $\mathbf{q_n}$, the score would be $s(r_i, q_n) = \mathbf{r_i}.\mathbf{q_n}$. The training objective is to minimize cross-entropy loss between the one-hot gold truth and the vector of predicted scores:

$$L(r_n, q_n) = -\log\left(\frac{\exp(s(r_n, q_n))}{\sum_i \exp(s(r_i, q_n))}\right)$$

We take the top one thousand relations from training data and use them as negative examples, excluding the gold. We compute the vector representation of all relations only once for each batch during training. Since relation representations are independent of the question representations, they can be reused for all examples in the batch. However, due to parameter update, they need to be computed anew for each batch.

Inference: During inference, we use $s(r, q)$ for scoring and ranking relations. Since the model parameters stay fixed, we compute the relation representations for all relations only once. If candidate KB relations are available from Entity analysis, we pick the highest-ranked relation from that set.

3 Evaluation

In this section, we detail our experimental setup and evaluate our approach against the state-of-the-art KBQA relation linking approaches. For fair comparisons, we replicate the same settings adopted by the systems we compare with both in terms of datasets and metrics.

3.1 Experimental Setup

Benchmarks: We perform experiments on four datasets targeting two popular KBs, DBpedia and

[2]The AMR parser of Astudillo et al. (2020) provides node to word alignments.

Wikidata. Each question in these datasets comes with its corresponding SPARQL query, annotated with gold relations. In particular, we used the following datasets:

- **QALD-9** (Usbeck et al., 2017): a dataset based on DBpedia with 150 test questions in natural language.
- **LC-QuAD 1.0** (Trivedi et al., 2017): another dataset based on DBpedia with a total of 5,000 questions (4,000 train and 1,000 test) based on templates.
- **LC-QuAD 2.0** (Dubey et al., 2019): A large dataset based on Wikidata with 6,046 test questions and around 24k training questions. Questions in this dataset have good variety and complexity levels such as multi-fact questions, temporal questions and questions that utilise qualifier information.
- **SimpleQuestions** (Diefenbach et al., 2017): A version of the popular SimpleQuestions dataset mapped to Wikidata. It comprises of 5,622 test questions, and around 19K training questions. As the name implies, all questions in this dataset are simple with queries encompassing a single triple in the KB.

Training: We train SemReL for DBpedia on the train data of LC-QuAD 1.0 and QALD-9. In addition, we use a subset of 80k examples from the distance supervisions data prepared by Mihindukulasooriya et al. (2020). This dataset is generated by retrieving Wikipedia sentences that contained pairs of entities from Knowledge Base triples. For our experiments, we filter our the sentences where the AMR path between the entities is more than two hops. For Wikidata experiments we train out system on the LC-QuAD 2.0 train dataset. Encoder parameters are initialized with the pretrained BERT base model (Wolf et al., 2020).

Baselines: For the DBpedia-based benchmarks, we compare SemReL with Falcon (Sakor et al., 2019a) and SLING (Mihindukulasooriya et al., 2020). As for Wikidata-based benchmarks, we compare against Falcon 2.0 (Sakor et al., 2020) and KB-Pearl (Lin et al., 2020).

[3]The KBPearl paper reports F1 of 0.41 due to a typo but its authors confirmed the correct F1 to be 0.52.

Dataset	Method	P	R	F1
QALD-9	Falcon	0.23	0.23	0.23
	SLING	0.39	0.50	0.44
	SemReL	0.46	0.44	**0.45**
LC-QuAD 1.0	Falcon	0.42	0.44	0.43
	SLING	0.41	0.55	0.47
	SemReL	0.51	0.51	**0.51**
LC-QuAD 2.0	Falcon 2.0	0.44	0.37	0.40
	SemReL	0.59	0.38	**0.46**
LC-QuAD 2.0 (1942 set)	KB-Pearl*	0.57	0.48	0.52[3]
	SemReL	0.70	0.45	**0.55**
Simple Questions	Falcon 2.0	0.35	0.44	0.39
	SemReL	0.69	0.70	**0.69**

Table 1: SemReL compared to SoTA systems on the DBpedia (above) and Wikidata (below) benchmarks.

Setup	all	one-hop	multi-hop
SemReL	0.51	0.54	0.50
w/o AMR	0.49	0.53	0.47
w/o TEXT	0.38	0.37	0.39
w/o KB rels	0.46	0.48	0.45

Table 2: SemReL F1 for *all*, *one-hop* and *multi-hop* questions with inputs ablated on LC-QuAD 1.0 testset. 'KB rels' refers to Knowledge Base relation candidates.

3.2 Results and Discussion

Table 1 compares SemReL with existing approaches. KB-Pearl used a subset of 1,942 test questions in their LC-QuAD 2.0 evaluation. For fair comparison, we also evaluate SemReL on the same subset.

SemReL outperforms all baselines across all benchmarks with respect to F1 score. Note that the baseline systems, specially SLING, achieve higher recall than precision. In contrast, SemReL has either balanced precision and recall, or much higher precision. This is in part due to missing entity or slot predictions, indicating that improving the preprocessing can further boost the system's performance. The results on SimpleQuestions are also worth noting, since the corresponding training set was not used in Wikidata model training. We also performed a zero-shot cross-KB experiment where we test our Wikidata model on a DBpedia dataset, LC-QuAD 1.0. The model is tested as is, and despite the relation names and granularity differences, it achieves an F1 of 0.33.

Ablation on Model Inputs: Table 2 shows the results of ablation experiments on LC-QuAD 1.0 testset where each of the system inputs are removed one at a time. As expected, the question text is the most crucial input: when combined with either KB candidates or AMR, it shows good performance. When AMR is removed, overall score drops by 2 points; it mostly comes from multi-hop questions. This indicates that focusing on different subgraphs of the input AMR improves retrieval of multi-hop relations. A similar effect was observed on QALD-9 and LC-QuAD 2.0 test sets when AMR was removed, degrading performance by 4.0 and 2.9 points respectively.

Impact on KBQA Performance: We integrated SemReL into the Neuro-Symbolic Question Answering (NSQA) system of Kapanipathi et al. (2021). NSQA is a modular system for KBQA where each sub-task is handled by a different module, allowing easy integration of new components. We found that the impact of using AMR in relation linking translates into nice performance gains in overall KBQA results. When AMR is incorporated in the relation linking module, the system performance on LC-QuAD 1.0 test dataset improves by 2.4 achieving a new state-of-the-art F1 of 44.5. We refer the reader to the NSQA paper (Kapanipathi et al., 2021) for more details on the system and experiments.

4 Related Work

Several relation linking systems have been proposed recently (Mulang et al., 2017; Singh et al., 2017; Dubey et al., 2018; Sakor et al., 2019a; Pan et al., 2019; Lin et al., 2020). Most of these methods are rule-based and rely solely on the question text and/or its dependency parse. Therefore, they try to improve their question understanding by using standard NLP tools such as POS tagging, tokenization n-gram tiling and even lexical database such as WordNet. FALCON 2.0 (Sakor et al., 2020) is a joint entity and relation linking tool over Wikidata. it uses a search engine indexed with Wikidata, a pipeline of text processing including POS tagging, tokenization, N-gram tiling/splitting and a catalog of rules for entity and relation linking. KBPearl (Lin et al., 2020) is another system that performs joint entity and relation linking to Wikidata. It first create a semantic graph of text using OpenIE and maps both entities and relations to a given KB.

However, none of the above mentioned methods for KBQA perform relation linking on two different KBs using the same system whilst our work is the first to perform relation linking over both DBpedia and Wikidata using the same system. In addition, some of these systems are KG-specific; e.g. Falcon (Sakor et al., 2019a) vs. Falcon 2.0 (Sakor et al., 2020), where adapting it from one KG to another requires non-trivial changes. Unlike these systems, SemReL leverages well-established semantic parsers such as AMR to achieve out-of-the-box better question representation.

Similar to our approach, SLING (Mihindukulasooriya et al., 2020) is a relation linking framework based on DBpedia which leverages semantic parsing using AMR and distant supervision. It consists of four distinct modules that capture different signals such as linguistic cues, semantic representation, and information from the knowledge base. Unlike SLING, SemReL is a KG-agnostic, single end-to-end neural model that does not require various ensemble components and yet achieves state-of-the-art performance on DBPedia and Wikidata datasets.

5 Conclusions and Future Work

In this paper, we present a simple transformer-based neural model for relation linking that leverages the semantic structure of a sentence. In contrast to existing systems such as SLING and Falcon, which are either rule-based or ensembles of several components, our neural architecture enables us to adapt the system to multiple KGs (e.g. DBpedia and Wikidata). It outperforms state-of-the-art systems on a variety of benchmarks.

Our ablation study shows that including the AMR graph improves performance, even with the relatively simple encoding scheme (plain text). In future, we will explore modeling the graph structure explicitly. This model also relies on a deterministic slot-finding algorithm based on AMR. While this identifies the correct relation slots most of the time, it is rule-based, and not always correct. In future work, we will explore learning algorithms to identify the slots from the AMR graph.

Finally, AMR parsers can be trained jointly with the relation linking objective. Currently, these parsers are sensitive to small changes in the input. Joint training can make them robust against text variations and more sensitive to the errors affecting slot identification and relation prediction.

References

Ramón Fernandez Astudillo, Miguel Ballesteros, Tahira Naseem, Austin Blodgett, and Radu Florian. 2020. Transition-based parsing with stack-transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, page 1001–1007.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dennis Diefenbach, Thomas Pellissier Tanon, Kamal Deep Singh, and Pierre Maret. 2017. Question answering benchmarks for wikidata. In *Proceedings of the ISWC 2017 Posters & Demonstrations and Industry Tracks co-located with 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 23rd - to - 25th, 2017.*

Mohnish Dubey, Debayan Banerjee, Abdelrahman Abdelkawi, and Jens Lehmann. 2019. LC-QuAD 2.0: A Large Dataset for Complex Question Answering over Wikidata and DBpedia. In *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part II*, volume 11779 of *Lecture Notes in Computer Science*, pages 69–78. Springer.

Mohnish Dubey, Debayan Banerjee, Debanjan Chaudhuri, and Jens Lehmann. 2018. Earl: joint entity and relation linking for question answering over knowledge graphs. In *ISWC*, pages 108–126.

Pavan Kapanipathi, Ibrahim Abdelaziz, Srinivas Ravishankar, Salim Roukos, Alexander Gray, Ramon Astudillo, Maria Chang, Cristina Cornelio, Saswati Dana, Achille Fokoue, et al. 2021. Leveraging abstract meaning representation for knowledge base-question answering. *Findings of the Association for Computational Linguistics: ACL 2021*.

Young-Suk Lee, Ramon Astuillo, Tahira Naseem, Revanth Reddy, Radu Florian, and Salim Roukos. 2020. Pushing the limits of amr parsing with self-learning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, page 3208–3214.

Jungwoo Lim, Dongsuk Oh, Yoonna Jang, Kisu Yang, and Heuiseok Lim. 2020. I know what you asked: Graph path learning using amr for commonsense reasoning. *arXiv preprint arXiv:2011.00766*.

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. *Proceedings of the VLDB Endowment*, 13(7):1035–1049.

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 *International Semantic Web Conference*, pages 402–419. Springer.

Isaiah Onando Mulang', Jennifer D'Souza, and Sören Auer. 2020. Fine-tuning BERT with focus words for explanation regeneration. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 125–130, Barcelona, Spain (Online). Association for Computational Linguistics.

Isaiah Onando Mulang, Kuldeep Singh, and Fabrizio Orlandi. 2017. Matching natural language relations to knowledge graph properties for question answering. In *SEMANTiCS 2017*, pages 89–96.

Jeff Z Pan, Mei Zhang, Kuldeep Singh, Frank van Harmelen, Jinguang Gu, and Zhi Zhang. 2019. Entity enabled relation linking. In *International Semantic Web Conference*, pages 523–538. Springer.

Ahmad Sakor, Isaiah Onando Mulang, Kuldeep Singh, Saeedeh Shekarpour, Maria Esther Vidal, Jens Lehmann, and Sören Auer. 2019a. Old is gold: linguistic driven approach for entity and relation linking of short text. In *NAACL: HLT 2019*, pages 2336–2346.

Ahmad Sakor, Isaiah Onando Mulang', Kuldeep Singh, Saeedeh Shekarpour, Maria Esther Vidal, Jens Lehmann, and Sören Auer. 2019b. Old is gold: Linguistic driven approach for entity and relation linking of short text. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2336–2346, Minneapolis, Minnesota. 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. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 3141–3148.

Kuldeep Singh, Isaiah Onando Mulang', Ioanna Lytra, Mohamad Yaser Jaradeh, Ahmad Sakor, Maria-Esther Vidal, Christoph Lange, and Sören Auer. 2017. Capturing knowledge in semantically-typed relational patterns to enhance relation linking. In *K-CAP 2017*, pages 1–8.

Priyansh Trivedi, Gaurav Maheshwari, Mohnish Dubey, and Jens Lehmann. 2017. Lc-quad: A corpus for complex question answering over knowledge graphs. In *ISWC 2017*, pages 210–218.

Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Bastian Haarmann, Anastasia Krithara, Michael Röder, and Giulio Napolitano. 2017. 7th open challenge on question answering over linked data (qald-7). In *Semantic Web Evaluation Challenge*, pages 59–69. Springer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

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.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2019. Zero-shot entity linking with dense entity retrieval. *ArXiv*, abs/1911.03814.