Answer Candidate Type Selection: Text-to-Text Language Model for Closed Book Question Answering Meets Knowledge Graphs

Mikhail Salnikov[1], **Maria Lysyuk**[1], **Pavel Braslavski**[3],
Anton Razzhigaev[1,2], **Valentin Malykh**[4], **Alexander Panchenko**[1,2]
[1]Skolkovo Institute of Science and Technology, [2]Artificial Intelligence Research Institute,
[3]Ural Federal University, [4]ISP RAS Research Center for Trusted Artificial Intelligence
{m.salnikov,a.panchenko}@skol.tech

Abstract

Pre-trained Text-to-Text Language Models (LMs), such as T5 or BART yield promising results in the Knowledge Graph Question Answering (KGQA) task. However, the capacity of the models is limited and the quality decreases for questions with less popular entities. In this paper, we present a novel approach which works on top of the pre-trained Text-to-Text QA system to address this issue. Our simple yet effective method performs filtering and re-ranking of generated candidates based on their types derived from Wikidata instance_of property. This study demonstrates the efficacy of our proposed methodology across three distinct one-hop KGQA datasets. Additionally, our approach yields results comparable to other existing specialized KGQA methods. In essence, this research endeavors to investigate the integration of closed-book Text-to-Text QA models and KGQA.

1 Introduction

Information stored in Knowledge Graphs (KG), such as Wikidata (Vrandecic and Krötzsch, 2014), for general domain or some specific knowledge graphs, e.g. for the medical domain (Huang et al., 2021), can be used to answer questions in natural language. Knowledge Graph Question Answering (KGQA) methods provide not a simple string as an answer, but instead an entity a KG.

Pre-trained Text-to-Text LMs, such as T5 (Raffel et al., 2019) or BART (Lewis et al., 2020), showed promising results on Question Answering (QA). Besides, recent studies have demonstrated the potential of Text-to-Text models to address Knowledge Graph Question Answering problems (Roberts et al., 2020; Sen et al., 2022).

While fine-tuning a Text-to-Text LM can significantly improve its performance, there are cases where questions cannot be answered without access to a knowledge graph, especially in case of less popular entities (Mallen et al., 2022): not all required knowledge can be "packed" into parameters of a neural model. However, even in such cases, Text-to-Text models can generate plausible answers that often belong to the *same type* as the correct answer. For example, Text-to-Text answers to the question "What is the place of birth of Philipp Apian?" are not correct (e.g., T5 model produced "Neuilly-sur-Seine" or "Freiburg im Breisgau" as answers), but these wrong candidates are of the correct type. Namely, the correct type "city" can be derived from the list of generated answers and used to perform a local KG search around the question entity "Philipp Apian" to derive the correct answer "Ingolstadt". Motivated by these observations, this study presents a method for answer type prediction utilizing the output of pre-trained Text-to-Text language models.

The contributions of our study are as follows: (1) A simple yet effective approach for improving generative KGQA using candidate answer type selection method based on instance_of properties aggregated from diversified beamsearch. (2) An open implementation of the method that is easily applicable to pre-trained generative models.[1]

2 Related Work

Traditional KGQA methods can be classified into two categories: retrieval-based and semantic parsing. Retrieval-based methods involve vectorizing the textual question and projecting it into a graph-based vector space containing candidate entities (Huang et al., 2019; Razzhigaev et al., 2023). Semantic parsing approaches generate formal question representations (e.g., SPARQL queries) to query a KG for the answer. Retrieval-based approaches rely on computationally expensive similarity searches using vector indices of millions of candidate entities. Semantic parsing requires maintaining a graph database capable of process-

[1]https://github.com/s-nlp/act

Proceedings of the 19th Conference on Natural Language Processing (KONVENS 2023), pages 155–164
September 18–22, 2023.

ing SPARQL queries.

Recently, to address these shortcomings of existing methods, a third wave of approaches emerged based on pre-trained Text-to-Text LMs such as T5 (Raffel et al., 2019) or BART (Lewis et al., 2020). Given a question, these models generate a label of the answer that can be directly linked to the entity in a KG. These models are more computationally convenient and they are described below.

The *Text-To-Text Transfer Transformer (T5)* (Raffel et al., 2019) is effective for question answering, as demonstrated by Roberts et al. (2020), or as part of a retrieval pipeline (Izacard and Grave, 2021). Furthermore, it has been shown that training T5 with Salient Span Masking (SSM) improves the model's performance on QA task. T5-ssm involves tuning T5 as a language model, masking *entities* instead of random tokens. T5-ssm-nq is a variant of the T5-ssm that is additionally fine-tuned on the NaturalQuestions (NQ) (Kwiatkowski et al., 2019) dataset. *BART*, a Text-to-Text model trained as a denoising autoencoder (Lewis et al., 2020), can also be applied to KGQA task (Cao et al., 2022).

3 Answer Candidate Type Selection

This section presents our proposed approach, Answer Candidate Type (ACT) Selection. We propose a universal approach to selecting the correct answer in the KGQA task by using any pre-trained sequence-to-sequence (seq2seq) model (in our cases a Text-to-Text Language Model) to generate answer candidates and to infer the type of expected answer. The answer candidate type selection pipeline shown in Figures 1 and 2 consists of four parts: the Text-to-Text model for candidate generation, Answer Type Extractor, Entity Linker, and the Candidate Scorer.

3.1 Initial Answer Candidate List Generation

To increase the diversity of the generated results, we use Diverse Beam Search (Vijayakumar et al., 2016) to generate an initial list of answer candidates C. It often leads to a better exploration of the search space by ensuring that alternative answers are considered. We define the types of entities using the Wikidata property `instance_of` (P31). Note that an entity can be of multiple types. Finally, the initial list of answer candidates is used in the Answer Candidate Typing and the Candidate Scorer with the mined candidates.

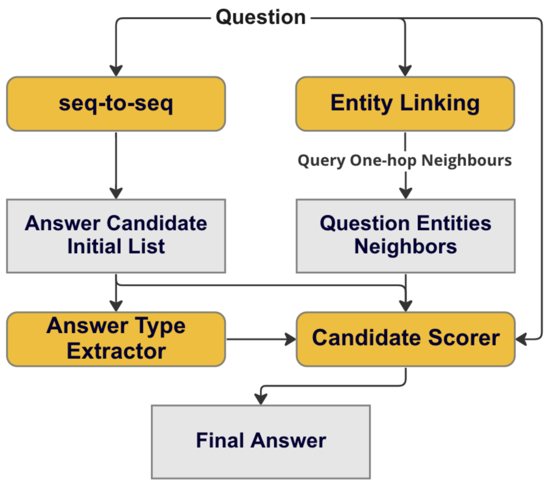

Figure 1: Answer Candidate Type (ACT) Selection.

3.2 Answer Candidate Typing

We rank all types by their frequency in the initial list of answer candidates. After that, we merge the top-K most frequent types and similar types to the final list T. Types similarity is calculated as a cosine similarity between Sentence-BERT (Reimers and Gurevych, 2019) embeddings of respective labels. The final types are defined as the ones where similarity is greater than a threshold.

A similar aggregation method using hypernyms (also known as "is-a" or "instance-of" relations) was used in the past to label clusters of words senses in distributional models (Biemann and Riedl, 2013; Panchenko et al., 2017): distributionally similar words share common hypernym and "top" common hypernyms are surprisingly good labels for sense clusters. The analogy in our method is that Text-to-Text models appear to produce a list of distributionally similar candidates.

3.3 Entity Linking

To enrich the list of candidates, we add all one-hop neighbours of the entities found in the question. For that we use the fine-tuned spaCy Named Entity Recognition (NER)[2] and the mGENRE (Cao et al., 2021) entity linking model.

3.4 Candidates Scorer

Finally, we calculate four scores for each answer candidate and rank them based on the weighted sum of the scores. The scores are as follows:

(1) Type score represents the size of the intersection between the set of types extracted from the

[2] `https://spacy.io`. More details about fine-tuning of the NER can be found in Appendix A.

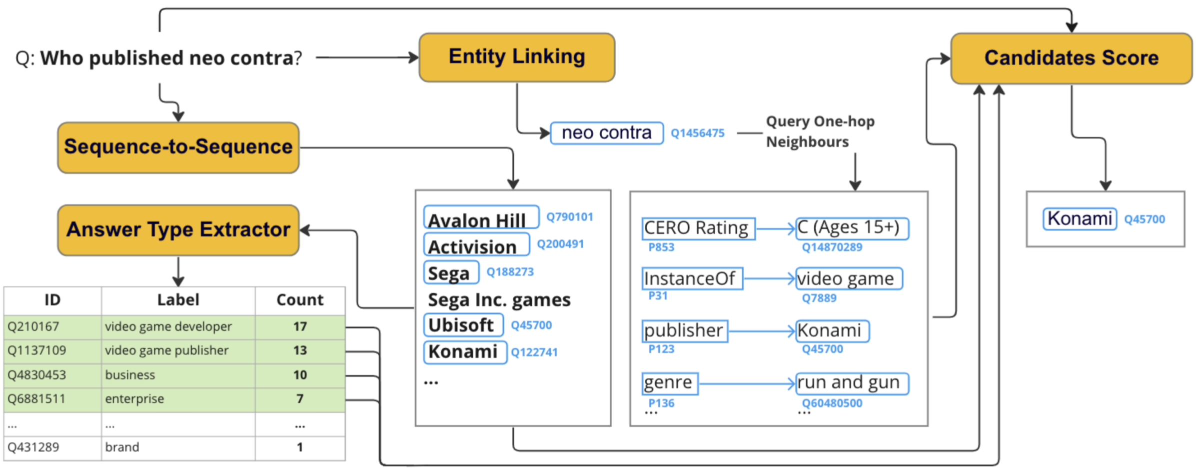

Figure 2: An example of the proposed Answer Candidate Type (ACT) Selection result.

answer candidates and the selected answer types. It is weighted by the number of selected answer types:

$$S_{\text{type}} = \frac{|\text{Candidates' Types} \cap T|}{|T|}.$$

(2) Forward one-hop neighbors score $S_{\text{neighbour}}$ is assigned 1 if the candidate is among the neighbors of the question entities, and 0 otherwise.

(3) Text-to-Text answer candidate score is determined by the rank of the candidate in the initial list C generated by the Text-to-Text model divided by the size of the list:

$$S_{\text{t2t}} = \frac{C.index(\text{Candidate})}{|C|}.$$

(4) Question-Property Similarity score S_{property} measures the cosine similarity between the embeddings of the relevant property and the entire question. We employ Sentence-BERT (Reimers and Gurevych, 2019) to encode the question, following a similar approach used for the Answer Candidate Type module.

The four scores are calculated for each entity and then are combined to generate a final score that determines the entity's ranking. The answer with the highest weighted sum of scores in the candidate list is selected as the final answer:

$$S_{\text{final}} = S_{\text{type}} + S_{\text{neighbour}} + S_{\text{t2t}} + S_{\text{property}}.$$

4 Experiments

We fine-tuned the Text-to-Text and spaCy NER models by using the entire training part of the respective datasets and fitting the model for eight epochs. The initial answer candidate lists were generated using Diverse Beam Search with 200 beams and a diversity penalty of 0.1. The Answer Candidate Typing module utilized the top-3 types and a similarity threshold of 0.6.

4.1 Data

We evaluate the ACT Selection on three Wikidata datasets containing one-hop questions. *SimpleQuestions-Wikidata (SQWD)* (Diefenbach et al., 2017) is a mapping of SimpleQuestions (Bordes et al., 2015) to Wikidata containing 21,957 questions. *RuBQ* (Korablinov and Braslavski, 2020; Rybin et al., 2021) is a KGQA dataset that contains 2,910 Russian questions of different types along with their English translations. *Mintaka* (Sen et al., 2022) is a multilingual KGQA dataset composed of 20,000 questions of different types. For our experiments we took only *generic* questions, whose entities are one hop away from the answers' entities in Wikidata, which resulted in 1,757 English questions.

4.2 Evaluation

We hypothesize that even if a closed-book QA text-to-text model returns an incorrect answer, the odds are that it is of the correct type.

The present study involves the extraction of answer types from Text-to-Text generated answers, followed by a comparison with the ground-truth answer types in the SQWD dataset. Our experimental findings demonstrate that the fine-tuned T5-Large-SSM model equipped with the ACT Selection can accurately predict the correct answer type in **94%** of the cases, while only **61%** of the candidate answers share the same type as the correct answer.

Model	SQWD	RuBQ en
QAnswer	33.31	32.30
KEQA TransE PTBG	**48.89**	33.80
ChatGPT	15.32	36.53
T5-Large-ssm (fine-tuned)	23.66	21.44
Ours: T5-Large-ssm (fine-tuned)	47.42	26.02
T5-11b-ssm-nq (zero-shot)	10.94	33.38
Ours: T5-11b-ssm-nq (zero-shot)	38.51	**38.31**

Table 1: Comparsion of the ACT Selection with KGQA baselines in terms of Hit@1 for SimpleQuestion-Wikidata (SQWD) with T5-Large-ssm fine-tuned on its training part and T5-11b-ssm-nq in zero-shot mode.

These results have provided an impetus to leverage this information to facilitate question-answering.

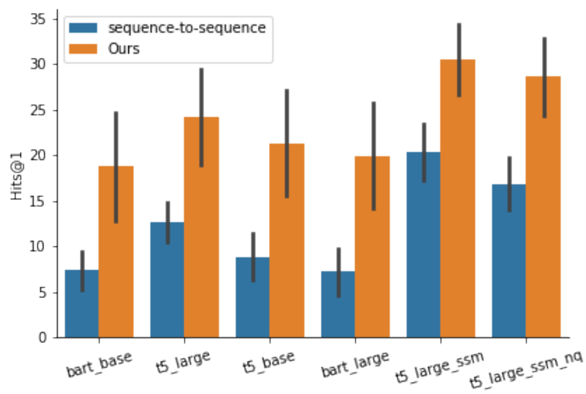

Figure 3: Average Hit@1 scores for the tuned models on SQWD, RuBQ, and Mintaka datasets from Table 2.

We evaluate the performance of two commonly used architecture types, T5 and BART. The proposed approach consistently improves the results of the Text-to-Text models on various datasets, as illustrated in Figure 3. We compare the mean Hit@1 scores of the tuned Text-to-Text models with the aforementioned datasets. Text-to-Text models were fine-tuned on the train splits of SQWD and the full train split of Mintaka datasets, and subsequently evaluated on the test splits of SQWD, RuBQ, and Mintaka using both tuned versions of the models.

As demonstrated in Table 2, the proposed approach consistently enhances the quality of KGQA tasks across various Text-to-Text models. Furthermore, we conducted experiments to verify that the proposed method can be employed with the Text-to-Text models in a zero-shot learning manner, without any fine-tuning. The benefits of the approach, in terms of quality improvement, are more noticeable when applied to smaller models. For example, the T5-large model, with its 737 million parameters, when paired with ACT Selection, delivers comparable performance to the T5-11b model, which has 11 billion parameters.

In line with expectations, larger models generally yield superior results. Notably, T5 models using the suggested method outperformed BART models. Moreover, across all tested T5 and BART models, implementing the ACT Selection markedly enhanced the performance of the foundational Text-to-Text model.

Table 1 showcases performance comparison between our suggested method and prominent KGQA systems, namely QAnswer (Diefenbach et al., 2020), KEQA (Huang et al., 2019), and chatGPT.[3] QAnswer is a multilingual rule-based system that tranforms the question into a SPARQL query. KEQA utilizes TransE embeddings of 200 dimensions, trained on Wikidata using the Pytorch-BigGraph (PTBG) framework (Lerer et al., 2019). ChatGPT is a conversational model that was launched in late 2022 and has received worldwide acclaim. Further details about evaluating ChatGPT and other generative models through entity-linked predictions can be found in appendix B. The tabulated data reveals that our approach delivers outcomes commensurate with those of state-of-the-art (SOTA) systems.

4.3 Ablation Study

We conducted an ablation study (cf. Table 3) to investigate the effects of the proposed scores on the candidate set collection process. Our main goal was to confirm that incorporating type information enhances candidate selection. We observed that methods relying solely on scores (such as Question-Property Similarity score) were not as effective as the ACT Selection approach.

Furthermore, we examined the necessity of initial candidates generated by the Text-to-Text model and whether restricting to question entity neighbors was sufficient. This investigation aimed to determine the added value of initial candidates in the selection process.

4.4 Error Analysis

We showed above that the ACT Selection approach fixed errors produced by the Text-to-Text LMs. We evaluate this approach using a subset of questions and predictions from the T5-Large-SSM model for the SQWD dataset. Our focus is on questions

[3] https://openai.com/blog/chatgpt

	SimpleQuestions-Wikidata			*RuBQ (English)*			*Mintaka (one-hop, English)*		
Tuned on →	**Zero-shot**	**SQWD**	**Mintaka**	**Zero-shot**	**SQWD**	**Mintaka**	**Zero-shot**	**SQWD**	**Mintaka**
BART-base	0	16.54	7.08	0	5.93	3.72	0	2.06	9.12
Ours	30.38	**42.60**	30.70	9.50	11.65	**11.72**	4.70	5.88	**10.29**
BART-large	0	16.97	3.02	0	4.07	4.86	0	1.76	12.65
Ours	30.42	**42.64**	31.39	9.50	12.15	**12.79**	4.41	5.29	**15.29**
T5-base	0	21.26	6.19	0	6.22	6.93	0	4.41	8.24
Ours	30.47	**43.13**	34.60	9.44	14.44	**16.58**	4.71	8.53	**10.59**
T5-large	0	22.36	9.43	0	11.15	12.15	0	7.06	14.41
Ours	29.88	**43.05**	36.89	9.44	18.94	**20.51**	4.71	10.00	**15.88**
T5-large-ssm	0.57	23.66	5.92	0.42	21.44	23.87	0.50	19.71	27.65
Ours	23.39	**47.42**	36.54	9.72	26.02	**27.88**	6.76	18.53	**28.24**
T5-large-ssm-nq	5.12	22.52	4.34	18.87	17.80	19.23	17.65	14.12	23.24
Ours	35.09	**43.88**	36.39	**27.52**	25.38	26.38	22.94	14.12	**25.59**
T5-11b-ssm	1.81	—	—	14.09	—	—	20.88	—	—
Ours	**25.84**	—	—	**20.94**	—	—	**24.71**	—	—
T5-11b-ssm-nq	10.94	—	—	33.38	—	—	41.76	—	—
Ours	**38.51**	—	—	**38.31**	—	—	**45.00**	—	—

Table 2: Evaluation results on three one-hop KGQA datasets (Hit@1 scores): comparing Text-To-Text Language Model with and without our proposed ACT Selection approach in zero-shot (without tuning for QA) or tuned on SQWD or Mintaka.

	Type score	Forward one-hop neighbours score	Text-to-Text LM candidates score	Question-Property Similarity score	All scores
Only initial candidates generated by Text-to-Text	2.51	31.73	27.04	31.82	35.89
Only question neighbours candidates	5.07	4.84	4.52	29.86	30.06
Full answer candidates set	2.81	5.46	27.04	30.75	**47.42**

Table 3: Ablation study of ACT Selection. Reporting Hit@1 at SQWD for T5-large-ssm fine-tuned on SQWD.

where the model's top-1 prediction was incorrect, but the ACT Selection approach extracted the correct answer.

The Text-to-Text model generated the correct answer in only 58.4% of questions in the chosen subset. However, our Entity Linking module was able to correctly extract 99.11% of question entities for this subset. The extraction of additional candidates from the question entity neighbors played a critical role in finding the correct answer.

5 Conclusion

We introduced a method for question answering over knowledge graph based on post-processing of beam-search outputs of a Text-to-Text model. Namely, a simple aggregation of KG "instance-of" relations is used to derive a likely type of the answer. This simple technique consistently improves performance of various Text-to-Text LMs favorably comparing to both specialized KGQA methods and ChatGPT with a carefully selected prompt and entity linked output on three distinct English one-hop KGQA datasets.

Our method may be also used to directly perform answer typing. In principle, it can be straightforwardly adapted to multilingual setup, but also multi-hop questions. We find it promising to use the method with larger pre-trained models to further boost performance as our current experiments show that the a quality growth as the model size increased.

6 Limitations

The main limitation of the current study is that the approach was only tested for one-hop questions. In principle, one can, however, sample candidates from graph from arbitrary subgraphs, e.g. second-order ego-networks of entity found in question. At the same time, improvements shown in this paper may not nessesarily generalize to such setting and need to be tested.

Another limitation is using diverse beam search, which is a computationally more expensive process as it requires larger beam sizes, usually.

Finally, requesting KG data can be a bottleneck if one is using a public SPARQL endpoint with query limits. This limitation can be alleviated by using an in-house private copy of a KG.

7 Ethical Considerations

Large pre-trained Text-to-Text models such as those used in our work are trained on datasets which may contain biased opinions. Therefore, QA/KGQA systems built on top of such models may transitively reflect such biases potentially generating stereotyped answers to the questions. As a consequence, it is recommended in production, not research settings, to use a special version of debiased pre-trained neural models and/or other technologies for the alleviation of the undesired biases of LLMs.

References

Chris Biemann and Martin Riedl. 2013. Text: now in 2d! A framework for lexical expansion with contextual similarity. *J. Lang. Model.*, 1(1):55–95.

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. *CoRR*, abs/1506.02075.

Nicola De Cao, Ledell Wu, Kashyap Popat, Mikel Artetxe, Naman Goyal, Mikhail Plekhanov, Luke Zettlemoyer, Nicola Cancedda, Sebastian Riedel, and Fabio Petroni. 2021. Multilingual autoregressive entity linking. *CoRR*, abs/2103.12528.

Shulin Cao, Jiaxin Shi, Liangming Pan, Lunyiu Nie, Yutong Xiang, Lei Hou, Juanzi Li, Bin He, and Hanwang Zhang. 2022. KQA Pro: A large diagnostic dataset for complex question answering over knowledge base. In *ACL'22*.

Dennis Diefenbach, Andreas Both, Kamal Singh, and Pierre Maret. 2020. Towards a question answering system over the semantic web. *Semantic Web*, 11(3):421–439.

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.*

Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge graph embedding based question answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*, pages 105–113. ACM.

Xiaofeng Huang, Jixin Zhang, Zisang Xu, Lu Ou, and Jianbin Tong. 2021. A knowledge graph based question answering method for medical domain. *PeerJ Computer Science*, 7:e667.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 874–880. Association for Computational Linguistics.

Vladislav Korablinov and Pavel Braslavski. 2020. Rubq: A russian dataset for question answering over wikidata. In *The Semantic Web - ISWC 2020 - 19th International Semantic Web Conference, Athens, Greece, November 2-6, 2020, Proceedings, Part II*, volume 12507 of *Lecture Notes in Computer Science*, pages 97–110. Springer.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Adam Lerer, Ledell Wu, Jiajun Shen, Timothée Lacroix, Luca Wehrstedt, Abhijit Bose, and Alex Peysakhovich. 2019. Pytorch-biggraph: A large scale graph embedding system. In *Proceedings of Machine Learning and Systems 2019, MLSys 2019, Stanford, CA, USA, March 31 - April 2, 2019*. mlsys.org.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Hannaneh Hajishirzi, and Daniel Khashabi. 2022. When not to trust language models: Investigating effectiveness and limitations of parametric and non-parametric memories. *CoRR*, abs/2212.10511.

Alexander Panchenko, Eugen Ruppert, Stefano Faralli, Simone Paolo Ponzetto, and Chris Biemann. 2017. Unsupervised does not mean uninterpretable: The case for word sense induction and disambiguation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 86–98, Valencia, Spain. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Anton Razzhigaev, Mikhail Salnikov, Valentin Malykh, Pavel Braslavski, and Alexander Panchenko. 2023. A system for answering simple questions in multiple languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 524–537, Toronto, Canada. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 5418–5426. Association for Computational Linguistics.

Ivan Rybin, Vladislav Korablinov, Pavel Efimov, and Pavel Braslavski. 2021. Rubq 2.0: An innovated russian question answering dataset. In *The Semantic Web - 18th International Conference, ESWC 2021, Virtual Event, June 6-10, 2021, Proceedings*, volume 12731 of *Lecture Notes in Computer Science*, pages 532–547. Springer.

Priyanka Sen, Alham Fikri Aji, and Amir Saffari. 2022. Mintaka: A complex, natural, and multilingual dataset for end-to-end question answering. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1604–1619, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Sowmya Vajjala and Ramya Balasubramaniam. 2022. What do we really know about state of the art ner? In *Proceedings of the Thirteenth Language Resources and Evaluation Conference, LREC 2022, Marseille, France, 20-25 June 2022*, pages 5983–5993. European Language Resources Association.

Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *CoRR*, abs/1610.02424.

Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

A Named Entity Recognition

According to the recent review of SOTA NER (Vajjala and Balasubramaniam, 2022), top-3 approaches were chosen: spaCy[4], Stanza[5] and SparkNLP[6]. Pre-rained NERs showed very poor quality ranging from 64% to 88% of missing cases for the SQWD data set. Among them, spaCy was the best; therefore, the standard spaCy configuration[7] was chosen for further fine-tuning. This pipeline requires two main pre-processing steps. First, the span of the entity should be fed into the algorithm. This span is predefined for Mintaka. However, for SQWD and RuBQ only Wikidata IDs of the entities are presented. Therefore, it was necessary first to define labels of the entities and all corresponding redirects. Next, these labels should have been found in the initial sentence for the span detection. Since for some of the entities there was no direct match in the sentence, the fuzzy search[8] was started. Second, spaCy requires the tag of the entity label (e.g., PERSON for Elon Musk , ORG for Tesla - the so-called BIO type tagging) for training, but in the initial data this label is missing. PERSON tag was chosen as the one for all cases. Additional experiments with partial data tagging (defining exact tag for each entity) were not successful.

B Evaluation generative models on KGQA problem

To link predicted answers with entities, we utilized the full-text search engine provided by the Wikidata API[9]. For answers generated by ChatGPT, we performed an additional step of removing the trailing dot at the end of the prediction (e.g., changing 'Yes.' to 'Yes'). For RuBQ dataset we just checked that predicted entity is one of the possible answers.

For predicting answers in the KGQA style, we experimented with different prompts for ChatGPT. Specifically, we used the prompt 'Answer as briefly as possible without additional information.' for evaluating the SQWD dataset and 'Answer as briefly as possible. The answer should be 'Yes', 'No' or a number if I am asking for a quantity of something, if possible, otherwise just a few words.'

[4] https://spacy.io
[5] https://stanfordnlp.github.io/stanza/
[6] https://nlp.johnsnowlabs.com
[7] https://spacy.io/usage/training/
[8] https://pypi.org/project/fuzzywuzzy/
[9] https://www.wikidata.org/w/api.php

for the RuBQ dataset.

C Examples

In this section, we include figures that illustrate examples of the working pipeline. Figure 2 presents the pipeline for the question "Who published neo contra?" The Text-to-Text model generates a set of answer candidates, such as "Avalon Hill," "Activision," and "Sega." These candidates are used to extract the type information, such as "video game developer." This type information is then employed in the Candidate Score module to rerank the final set of candidates, ultimately identifying the correct answer as "Konami."

Additionally, in Figures 4, 5, and 6, we provide additional examples that demonstrate the extraction of types and the calculation of scores within the pipeline.

Question: The champions of what two leagues played in the first four Super Bowls?
Target: Entity: Q1215884 (National Football League) (InstanceOf: Q15991290 (professional sports league))
Target: Entity: Q464508 (American Football League) (InstanceOf: Q623109 (sports league))

Final answers

Property	P Label	Entity	E Label	InstanceOf	instance of score	forward one hop neighbors score	answers candidates score	property question intersection score
		Q370883	National Football League	Q15991303 (association football league)	0.83333	0.00000	1.00000	0.00000
		Q464508	American Football League	Q623109 (sports league)	0.83333	0.00000	0.95455	0.00000
		Q190618	New York Giants	Q17156793 (American football team)	0.83333	0.00000	0.86364	0.00000
		Q213837	Green Bay Packers	Q17156793 (American football team)	0.83333	0.00000	0.72727	0.00000
		Q337758	San Francisco 49ers	Q17156793 (American football team)	0.83333	0.00000	0.68182	0.00000
		Q205033	Chicago Bears	Q17156793 (American football team)	0.83333	0.00000	0.59091	0.00000
		Q1784597	NFC Championship Game	Q13406554 (sports competition)	0.83333	0.00000	0.54545	0.00000
		Q193390	New England Patriots	Q17156793 (American football team)	0.83333	0.00000	0.50000	0.00000
		Q191477	Pittsburgh Steelers	Q17156793 (American football team)	0.83333	0.00000	0.40909	0.00000
		Q4743798	American Football Association	Q61718902 (Former association football federation)	0.83333	0.00000	0.36364	0.00000
		Q219714	Philadelphia Eagles	Q17156793 (American football team)	0.83333	0.00000	0.31818	0.00000
		Q594428	NFC East	Q3032333 (sports division)	0.83333	0.00000	0.27273	0.00000
		Q223527	Cleveland Browns	Q17156793 (American football team)	0.83333	0.00000	0.22727	0.00000
		Q238240	Eastern Conference	Q13406554 (sports competition)	0.83333	0.00000	0.09091	0.00000

Answers instanceOf count (selected)

InstanceOf	Label	Count
Q17156793	American football team	8.0
Q13406554	sports competition	2.0
Q15991303	association football league	1.0
Q623109	sports league	1.0
Q512187	federal republic	1.0
Q1489259	superpower	1.0
Q1520223	constitutional republic	1.0
Q3624078	sovereign state	1.0
Q5255892	democratic republic	1.0
Q6256	country	1.0
Q61718902	Former association football federation	1.0
Q3032333	sports division	1.0
Q67476316	college athletic conference	1.0
Q103495	world war	1.0
Q11514315	historical period	1.0
Q215380	musical group	1.0

Seq2Seq answers candidates

Entity	E Label	InstanceOf
Q370883	National Football League	Q15991303 (association football league)
Q464508	American Football League	Q623109 (sports league)
Q443821	NFL	Q4167410 (Wikimedia disambiguation page)
Q190618	New York Giants	Q17156793 (American football team)
Q30	United States of America	Q512187 (federal republic) Q1489259 (superpower) Q1520223 (constitutional republic) Q3624078 (sovereign state) Q5255892 (democratic republic) Q6256 (country)
Q225804	AFL	Q4167410 (Wikimedia disambiguation page)
Q213837	Green Bay Packers	Q17156793 (American football team)
Q337758	San Francisco 49ers	Q17156793 (American football team)
Q4649857	AAFC	Q4167410 (Wikimedia disambiguation page)
Q205033	Chicago Bears	Q17156793 (American football team)
Q1784597	NFC Championship Game	Q13406554 (sports competition)
Q193390	New England Patriots	Q17156793 (American football team)

kbqa_dev (salnikov_pg) @ nlp1 ⊗ 0 ⚠ 0 3 DVC (Auto) Jupyter Server: Local Cell 8 of 29

Figure 4: Example question: The champions of what two leagues played in the first four Super Bowls?

Question: who published neo contra?
Target: Entity: Q45700 (Konami) (InstanceOf: Q210167 (video game developer); Q219577 (holding company); Q891723 (public company); Q1137109 (video game publisher))

Final answers

Property	P Label	Entity	E Label	InstanceOf	instance of score	forward one hop neighbors score	answers candidates score	property question intersection score
P123	publisher	Q45700	Konami	Q210167 (video game developer) Q219577 (holding company) Q891723 (public company) Q1137109 (video game publisher)	0.69231	1.00000	0.63333	0.62404
P178	developer	Q45700	Konami	Q210167 (video game developer) Q219577 (holding company) Q891723 (public company) Q1137109 (video game publisher)	0.69231	1.00000	0.63333	0.59095
		Q652421	MicroProse	Q210167 (video game developer)	0.92308	0.00000	0.86667	0.00000
		Q790101	Avalon Hill	Q3579158 (board game publishing company) Q4830453 (business) Q100271038 (tabletop role-playing game publisher)	0.76923	0.00000	1.00000	0.00000
		Q200491	Activision	Q210167 (video game developer) Q658255 (subsidiary) Q1137109 (video game publisher)	0.76923	0.00000	0.96667	0.00000
		Q173941	Electronic Arts	Q891723 (public company) Q1137109 (video game publisher)	0.84615	0.00000	0.83333	0.00000
		Q660990	Avalanche Software	Q210167 (video game developer) Q4830453 (business)	0.84615	0.00000	0.80000	0.00000
				Q210167 (video				

Answers instanceOf count (selected)

InstanceOf	Label	Count
Q210167	video game developer	17.0
Q1137109	video game publisher	13.0
Q4830453	business	10.0
Q6881511	enterprise	7.0
Q891723	public company	7.0
Q100271038	tabletop role-playing game publisher	2.0
Q3579158	board game publishing company	1.0
Q658255	subsidiary	1.0
Q43229	organization	1.0
Q219577	holding company	1.0
Q507619	retail chain	1.0
Q726870	brick and mortar	1.0
Q18388277	technology company	1.0
Q431289	brand	1.0
Q1058914	software company	1.0

Seq2Seq answers candidates

Entity	E Label	InstanceOf
Q790101	Avalon Hill	Q3579158 (board game publishing company) Q4830453 (business) Q100271038 (tabletop role-playing game publisher)
Q200491	Activision	Q210167 (video game developer) Q658255 (subsidiary) Q1137109 (video game publisher)
Q122741	Sega	Q210167 (video game developer) Q1137109 (video game publisher) Q4830453 (business) Q6881511 (enterprise)
Q188273	Ubisoft	Q210167 (video game developer) Q891723 (public company) Q1137109 (video game publisher) Q43229 (organization)
Q652421	MicroProse	Q210167 (video game developer)
Q173941	Electronic Arts	Q891723 (public company) Q1137109 (video game publisher)
Q660990	Avalanche Software	Q210167 (video game developer) Q4830453 (business)
Q339228	Acclaim Entertainment	Q4830453 (business) Q6881511 (enterprise)

bqa_dev (salnikov_pg) @ nlp1 ⊗ 0 ⚠ 0 3 DVC (Auto) Jupyter Server: Local

Figure 5: Example question: Who published neo contra?

Question: what is the place of birth of sam edwards??
Target: Entity: Q23051 (Swansea) (InstanceOf: Q1549591 (big city); Q515 (city))

Final answers

Property	P Label	Entity	E Label	InstanceOf	instance of score	forward one hop neighbors score	answers candidates score	property question intersection score
P19	place of birth	Q23051	Swansea	Q1549591 (big city) Q515 (city)	0.93333	1.00000	0.00000	0.72962
P19	place of birth	Q219656	Macon	Q62049 (county seat) Q486972 (human settlement) Q1093829 (city in the United States) Q1549591 (big city) Q3301053 (consolidated city-county) Q76514543 (municipality of Georgia)	0.93333	1.00000	0.00000	0.72962
P20	place of death	Q1012665	Durango	Q62049 (county seat) Q1093829 (city in the United States)	0.96667	1.00000	0.00000	0.35553
P937	work location	Q350	Cambridge	Q1187811 (college town) Q1357964 (county town) Q1549591 (big city) Q515 (city)	0.93333	1.00000	0.00000	0.38336
P20	place of death	Q350	Cambridge	Q1187811 (college town) Q1357964 (county town) Q1549591 (big city) Q515 (city)	0.93333	1.00000	0.00000	0.35553
		Q126269	Wolverhampton	Q515 (city)	0.96667	0.00000	0.96923	0.00000
		Q205679	London Borough of Hackney	Q211690 (London borough) Q7897276 (unparished area)	0.93333	0.00000	1.00000	0.00000

Answers instanceOf count (selected)

InstanceOf	Label	Count
Q1549591	big city	16.0
Q515	city	13.0
Q7897276	unparished area	12.0
Q3957	town	12.0
Q1093829	city in the United States	11.0
Q18511725	market town	8.0
Q211690	London borough	4.0
Q1115575	civil parish	4.0
Q1637706	million city	4.0
Q62049	county seat	4.0
Q1357964	county town	3.0
Q188509	suburb	3.0
Q174844	megacity	2.0
Q200250	metropolis	2.0
Q208511	global city	2.0
Q2264924	port settlement	2.0
Q5119	capital city	2.0
Q13218391	charter city	2.0
Q2154459	New England town	2.0
Q748198	gay village	2.0
Q2755753	area of London	2.0
Q1074523	planned community	1.0
Q10270157	new town	1.0
Q15063611	city in the state of New York	1.0
Q51929311	largest city	1.0
Q15210668	lower-tier municipality	1.0
Q44551483	city in Newfoundland and Labrador	1.0
Q15221310	second-class city	1.0
Q6489113	large burgh	1.0
Q50330360	second largest city	1.0
Q745456	business cluster	1.0
Q106646149	Climate emergency declarations in New Zealand	1.0
Q3184121	municipality of Brazil	1.0
Q2974552	city in New Jersey	1.0

Figure 6: Example question: What is the place of birth of Sam Edwards?