The Combination of BERT and Data Oversampling for Relation Set Prediction

Thang Ta Hoang[1,2][0000−0003−0321−5106], Sabur Butt[1][0000−0002−4056−8923], Jason Angel[1][0000−0002−7991−1979], Grigori Sidorov[1][0000−0003−3901−3522], and Alexander Gelbukh[1][0000−0001−7845−9039]

[1] Instituto Politécnico Nacional (IPN), Centro de Investigación en Computación (CIC), Mexico City, Mexico
tahoangthang@gmail.com, sabur@nlp.cic.ipn.mx, ajason08@gmail.com, sidorov@cic.ipn.mx, gelbukh@cic.ipn.mx

[2] Dalat University, Lam Dong, Vietnam
thangth@dlu.edu.vn

Abstract. In this paper, we engage the Task 2 of the SMART Task 2021 challenge in predicting relations used to identify the correct answer of a given question. This is a subtask of Knowledge Base Question Answering (KBQA) and offers valuable insights for the development of KBQA systems. We introduce our method, combining BERT and data oversampling with text replacements of linked terms to Wikidata and dependent noun phrases, in predicting answer relations in two datasets. For the DBpedia dataset, we obtain F1 of 83.15%, precision of 83.68%, and recall of 82.95%. Meanwhile, for the Wikidata dataset we achieved F1 of 60.70%, precision of 61.63%, and recall of 61.10%.

Keywords: Knowledge Base Question Answering · Relation Prediction · Relation Linking · Semantic Web Challenge · ISWC

1 Introduction

In Natural Language Processing (NLP), Knowledge Base Question Answering (KBQA) is a task that deals with answering questions using the relevant information provided in the knowledge base (KB). Natural language questions are converted into SPARQL queries to retrieve answers from KBs. The question types can vary, depending on the targeted problem to find the answers, such as simple questions need to have small snippets of text, complex questions require inferencing and synthesizing information, or long questions which are more difficult to interpret, etc. NLP researchers normally build pre-defined templates to generate questions or use crowdsourcing to produce the desired questions.

To correctly map questions to relevant KB relations, relation linking is an important task for improving significantly the performance of question answering. It has been a challenging problem for NLP researchers due to multiple and implicit relations in questions, limited annotated training data, and lexical-semantic differences [25]. There is a dearth of methods and studies exploring

relation linking on available KBs. The current systems also fall short [25] in understanding the implicit relations or relations with lexical gaps. Additionally, the number of candidate relations in KBs can cause problems as well, if the text does not apply which relation should be preferred over the other. Besides, the benefits of relation linking can also be applied on social media text-based questions or social media question-answering in general [6].

In this paper, we participate in Task 2 – relation set prediction of the SMART Task 2021 [3] over DBpedia and Wikidata datasets [17]. For each question, our duty is to search for relations used to predict the correct answer. Each relation also consists of a list of candidate ontologies ranked by the relevance. Table 1 shows some examples extracted from the DBpedia and Wikidata datasets. The number of relations could be either 1, 2, or 3. In the DBpedia dataset, prefixes `dbo` and `dbp` mean "DBpedia ontology" and "DBpedia property", while `P582` means a Wikidata property (P for property) with identifier 582 in the Wikidata dataset.

Table 1. Some examples in DBpedia and Wikidata datasets.

Question	**Relation**	**Dataset**
Whats the birthplace of hans dally	relation1: - dbp:birthPlace - dbo:birthPlace	DBpedia
What are the awards won by the film director of Saraband ?	relation1: - dbo:director relation2: - dbp:awards	DBpedia
Was Herbert Marcuse an economist?	relation1: - P101 (field of work)	Wikidata
On what military branch Sigmund Jähn served until 1965?	relation1: - P241 (military branch) relation2: - P582 (end time)	Wikidata

We applied a combination of BERT models and data oversampling by text replacements of linked terms to Wikidata and dependent noun phrases to solve the problem. Besides this section, the other sections follow this structure: Section 2 explain the background of KBQA and techniques used to improve the problems in the field. Section 3 and Section 4 describe the datasets and our methodology used to train the models and produce the results. Finally, we present our experiments and error reports, as well as conclusions and future works, in Sections 5 and Section 6.

[3] https://smart-task.github.io/2021/

2 Literature Review

The task of relation set prediction requires a thorough understanding of the KBQA background. Question answering has evolved from simple QA by achieving significant results to complex QA tasks. Some popular datasets related to the task are Question Answering over Linked Data (QALD) [19], LC-QuAD [30], WebQuestions [3], ComplexQuestions [1], ComplexWebQuestions [29], WebQuestionsSP [35], and LC-QuAD 2.0 [10]. WebQuestions is built around real questions derived from the Google Suggest API, while QALD and LC-QuAD and are powered by DBPedia. Meanwhile, LC-Quad 2.0 comprises both DBPedia and Wikidata containing complex questions generated through SPARQL queries filled with associated relations and seed entities.

There have been several methods proposed for complex question answering that can be listed as Information Retrieval (IR) based methods, Neural Semantic Parsing based methods, and traditional methods. Traditional methods mainly rely on template-based models [2], whereas IR-based methods have included feature engineering (question word, focus word, topic word, central verb etc.) [33] and representation learning techniques (semantic matching in vector space, multi-hop reasoning) [4, 5]. On the majority of occasions, we have seen neural-based methods to lead the problem with techniques such as Encoder-Decoder [9, 32] and Query Graphs [24, 34] methods. For in-depth analyses on the existing techniques, we recommend referring to the study [15].

Some previous approaches to identify relation linking has been using semantic parsing [18] or hand-coded rules [26]. Autoregressive seq2seq models have proven to be effective in the past for problems like entity linking [8], question answering [16] or slot filling [22]. However, they need further attention for the problem of relation linking [25]. The closer approach to relation linking is GenrRL [25], a generative model for relation linking using pretrained seq2seq (BART) models for KBQA.

KBQA corpora are usually imbalanced due to they consists of numerous natural language questions, created from language diversity and human creativity. To help the dataset more balanced or less biased, oversampling and undersampling techniques are usually applied to reduce popular data and increase rare data. For oversampling, there are many techniques such as SMOTE [7], ADASYN [13], and data augmentation (EDA [31], GenAug [11], contextual augmentation [14] for text). In this paper, a simple oversampling technique based on text replacements is used to increase the number of questions, including rare ones. In questions, we replace the dependent noun phrases by their roots and linked terms to Wikidata (extracted by TagMe [12]) by their aliases. Cross-lingual data augmentation is also helpful for producing more new questions in different languages [28], but we do not apply here.

Referring to some studies [20, 21, 27] in the last year's challenge, we found that BERT outperformed other methods in predicting answer categories and types. Hence, we decided to choose BERT to examine how well BERT can go with the relation set prediction.

3 Dataset Analysis

From DBpedia and Wikidata datasets provided by organizers, we did some analysis before proceeding to the next steps. Firstly, we analyzed that the number of relations we had from given questions. The number of relations over questions is either 1, 2, or 3 as in Figure 1. Especially, the DBpedia dataset contains only 7 questions with 3 relations, while most of the questions will have only 1 relation. In the Wikidata dataset, questions with 2 relations take the biggest part, and questions with 3 relations have the least number of questions, but not too rare as those in the DBpedia dataset.

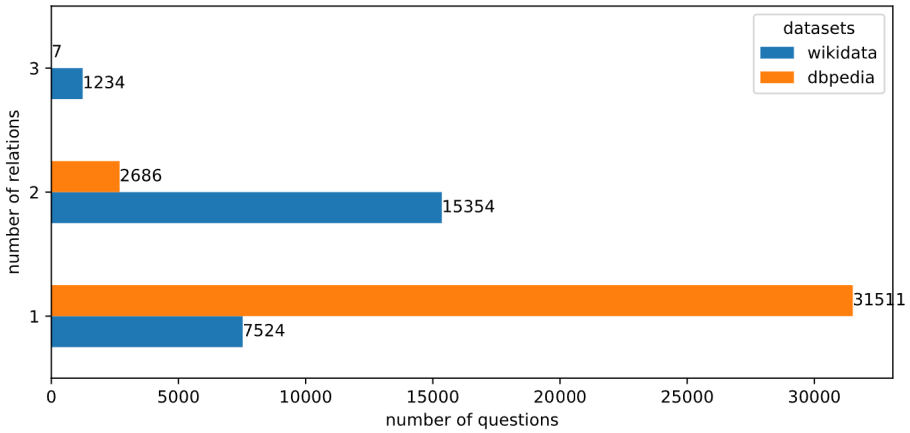

Fig. 1. The distribution of questions by relations over 2 datasets: Wikidata and DBpedia.

Next, for each question, we split its relation list into single relations, then form the distribution of relations over questions as in Figure 2. If the number of relations that appear are in less than 5 questions, we call them rare relations. We have 226 and 2299 ones corresponding to DBpedia and Wikidata datasets. It is clear that the datasets are imbalanced and contain many rare relations, thus this becomes a challenge for not only this paper but also for text classification.

We first thought about reducing the number of relations by depending on the ontology hierarchy structure or removing all rare types. However, the former takes time to analyze and the latter might affect the outcome performance in general. Therefore, we decided to apply an oversampling technique by text replacements of linked terms to Wikidata and dependent noun phrases to reduce the number of rare types as many as possible.

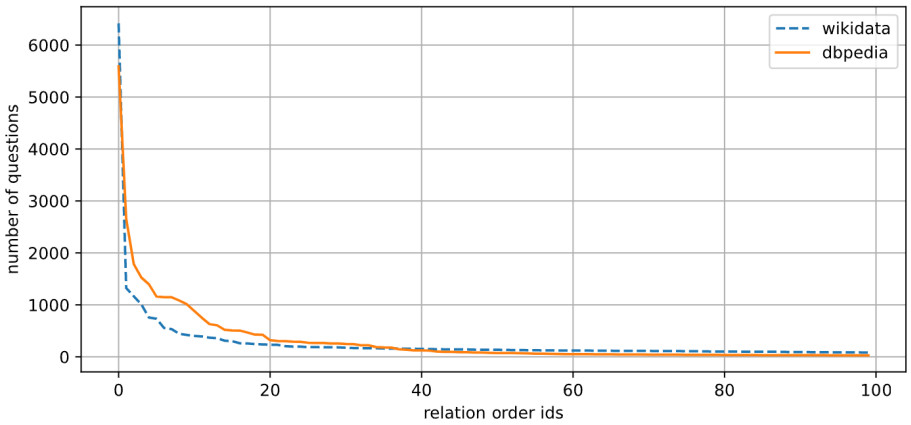

Fig. 2. The distribution of first 100 (single) relations ordered by the number of questions over 2 datasets: Wikidata and DBpedia

4 Methodology

4.1 Preprocessing and Oversampling methods

For each question, we used spaCy[4] v.2.3.2 to analyze the question structure and to get its components, such as question type, subject, main verb (also `ROOT`), terms (noun phrases, dependent noun phrases) in order to build the sentence template, and apply entity linking (EL) methods to extract terms connecting to Wikidata. We take some spaCy components such as `en_core_web_lg`, `STOP_WORDS`, `lemmatizer` (`Lemmatizer`, `ADJ`, `NOUN`, `VERB`), and `sentencizer` pipeline. The sentence template is built by a greedy algorithm which absorbs all longest terms. The text below (in `json` format) represents the structure analysis of a random example.

```
{
"question": "What is the safety classification and labeling for
↪  hydrochloric acid?",
"relations": ["P4952"],
"relation_labels": ["safety classification and labelling"],
"question_template": "What is {the safety classification} and
↪  {labeling} for {hydrochloric acid}?",
"key_terms": ["labeling", "hydrochloric acid"],
"subject": ["the safety classification"],
"main verb": ["is", -1],
"aux_verb": ["is", 1],
"entities": ["labeling", "hydrochloric acid", "acid"],
```

[4] https://spacy.io/

```
"question_type": "what",
"dependency_nouns": ["hydrochloric acid", "hydrochloride",
↪  "acid",...],
"el_terms": [ "hydrochloric acid": { "wikidata_id": "Q2409",
↪  "label": "Hydrochloric acid", "aliases": ["HCl", "muriatic
↪  acid"},...],
...
}
```


We found some terms containing typos and started to think how to correct them by using a grammar model. However, this approach may create more extra works. Instead, we deploy a simple method, API searching [5] of Wikipedia. For each mapped term no matter it has typos or not and longer than 8 characters, we used the API searching to fix typos may have. If the new term is the same with the original term, there is no need fix anything here. Otherwise, if the length of new term is equal or larger 1 or 2 than the old one, we will get this term. We assume the longer phrases can keep the original meaning better. However, this method is not always stable when the result could be a term that is more popular than the term we want.

For the EL, we already built APIs [6] for doing EL in several other methods, such as Babelfy, OpenTapioca, Wikifier, and AIDA but decided to use TagMe API due to its availability on D4Science.org [7]. Also on the same website, WAT API [8] is a better method based on TagMe but works only with English [23]. In the experiment, we found that the WAT API is not good as TagMe, so we consider it is an alternative solution. We gathered mapped terms to Wikidata which have a link probability higher than 0.9 to guarantee credibility. After that, we applied an oversampling technique over mapped terms to increase the discrepancy of questions over rare types. We simply replaced the mapped terms (its tokens) with their corresponding `aliases` to create new questions, producing new questions such as:


```
{
"question": "What is the safety classification and labeling for
↪  hydrochloric acid?",
"new questions": ["What is the safety classification and labeling
↪  for HCl?", "What is the safety classification and labeling for
↪  muriatic acid?",...],
...
}
```


In fact, we even produced more questions by replacing key terms by their roots, such as `safety classification` to `classification`. We assume these

[5] https://www.mediawiki.org/wiki/API:Search
[6] https://github.com/thangth1102/SMART_2021_Task2/tree/main/entity_linking
[7] https://sobigdata.d4science.org/web/tagme/tagme-help
[8] https://sobigdata.d4science.org/web/tagme/wat-api

Table 2. The comparisons between original and extended datasets. There are 695 relations in DBpedia datasets and 3171 relations in Wikidata datasets. The number of rare relations is counted for any relation appearing in less than 5 questions.

	Original datasets		Extended datasets	
	Questions	**Rare relations**	**Questions**	**Rare relations**
DBpedia	34204	226	77280	121
Wikidata	24112	2299	106729	1697

text replacements can help the models to deal better with the new data, since modifiers are used to carry less lexical information than headers of phrases. Table 2 shows the changes between the original datasets and extended datasets by our oversampling method. Compare to the original datasets, the new DBpedia dataset has about 2 times more questions and less than a roughly half rare relation. Meanwhile, about 5 times more questions and less than roughly a quarter of rare relations are results in the Wikidata dataset.

4.2 Training model

We applied `bert-based-cased` as a pretrained BERT model for the training process to see how well BERT can deal with relation set prediction from input questions. Considering relation set prediction is a problem of text classification, we thus flatten relation lists into strings in both DBpedia and Wikidata datasets to easier train. For example, the relation list `[['dbo:director'],['dbp:awards']]` of the question `"What are the awards won by the film director of Saraband ?"` will be flattened as string `dbo:director;dbp:awards`. The delimiter "`;`" refers to a divider between two relations.

Table 3. The statistics of DBpedia and Wikidata datasets after flattening relation lists. The number of rare string relations is counted for any string relation appearing in less than 5 questions.

Dataset	**Questions**	**String relations**	**Rare string relations**	**Questions per relation**
DBpedia	77280	2730	1834	28.30
Wikidata	106729	8416	4885	12.68

Table 3 indicates the statistics of flatten relations over DBpedia and Wikidata datasets. We only take the first item of each relation list for the flattening step to reduce the number of produced relation strings. For clear, this means each question has only one relation. As far as we know, according to the evaluation code [9], this may affect the final performance in general. However, there is

[9] https://github.com/smart-task/smart-2021-dataset/blob/main/evaluation/RL/evaluator.py

no guarantee that the performance is better if we keep all relation lists for the training. We will clarify this in the future works. Compare to Table 2, the numbers of relations and rare relations now change significantly, which hint us about the difficulty that models have to deal with.

5 Experiments and Error reports

For each flattened dataset, we split it into 3 subsets, training, validation, and test sets with the ratio 8:1:1. Within 15 epochs, the best model will be saved with the highest validation accuracy. In the Wikidata dataset, the training values are lower than those in the DBpedia dataset because it contains more relations. In Table 4, the validation accuracy of 0.84 is acceptable, but it suggests us the need to train the model more. After the training process, we validated our models with test and golden label sets provided by organizers to have final evaluations. For each dataset, the average values of precision, recall, and F1 metrics are applied to all questions.

Table 4. The training results of flatten datasets for relation set prediction.

Dataset	**Train acc**	**Test acc**	**Val acc**
DBpedia1	0.98	0.97	0.90
Wikidata	0.89	0.90	0.84

Table 5. The evaluation metrics by participant groups on the test set, which was provided by the organizers.

	DBpedia			Wikipedia		
Team	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Nadine	0.8613	0.8760	0.8623	0.7509	0.8163	0.7601
Our team	0.8368	0.8295	0.8315	0.6163	0.6110	0.6070

Table 5 shows our performance on the ranked table offered by organizers. Unfortunately, there are only 2 teams participating in the challenge. Compare to the other team, we have similar results on the DBpedia dataset, while on the Wikipedia dataset, we have a lower performance. This may be from a gap between the validation accuracy (0.84) and the training accuracy (0.89), which are also not expected scores in our training process.

In future, we should use all data in the training process instead of splitting into different sets, or train the model until meeting the smallest gap between accuracies in all sets. The performance of both teams reconfirms the task difficulty as declared by the organizers.

We see some minor errors in the dataset, but they do not affect the outcome performance in general. However, our text replacement method contains an error. The question `"What is the Beethoven's piano sonatas?"` will produce a new question as `"What is the Beethovensonatas?"` when replacing `'s piano sonatas` to its root `sonatas`. Therefore, we have to avoid all replacements on the possessive nouns containing `'s`. Besides, we will improve our parsing analysis because we can not extract the correct components from sentences in some cases.

6 Conclusion

In this paper, we participate in Task 2 of the SMART 2021 Semantic Web Challenge, relation set prediction. We applied spaCy and TagMe to extract sentence components and linked terms from questions. By using a simple oversampling method based on text replacements of linked terms to Wikidata and dependent nouns, we were able to expand the size of datasets, targeting to have a higher number of questions as many as possible, especially on rare answer relations.

In the experiments, a pre-trained BERT model, `bert-base-cased` is used for the training process on flatten datasets to predict relations. For the DBpedia dataset, precision and recall are 83.68% and 82.95% while F1 is 83.15%. We obtained lower metric values for the Wikidata dataset with the precision of 61.63%, recall of 61.10%, and F1 of 60.70%.

In the future, we will improve the analysis parsing of question structure and EL methods to add ontology information on top of the training data. We will also try with other neural networks or any hybrid approach to search for a better method, as well as try to augment the dataset by other entity linking methods and multilingual translation. At last, the semantic relationships between relations should be studied in linking to questions to minimize the number of relations and infer relations effectively.

Acknowledgements

The work was done with partial support from the Mexican Government through the grant A1-S-47854 of the CONACYT, Mexico, grants 20211784, 20211884, and 20211178 of the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico. The authors thank the CONACYT for the computing resources brought to them through the Plataforma de Aprendizaje Profundo para Tecnologías del Lenguaje of the Laboratorio de Supercómputo of the INAOE, Mexico and acknowledge the support of Microsoft through the Microsoft Latin America PhD Award.

References

1. Bao, J., Duan, N., Yan, Z., Zhou, M., Zhao, T.: Constraint-based question answering with knowledge graph. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. pp. 2503–2514 (2016)

2. Bast, H., Haussmann, E.: More accurate question answering on freebase. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. pp. 1431–1440 (2015)
3. Berant, J., Chou, A., Frostig, R., Liang, P.: Semantic parsing on freebase from question-answer pairs. In: Proceedings of the 2013 conference on empirical methods in natural language processing. pp. 1533–1544 (2013)
4. Bordes, A., Chopra, S., Weston, J.: Question answering with subgraph embeddings. arXiv preprint arXiv:1406.3676 (2014)
5. Bordes, A., Usunier, N., Chopra, S., Weston, J.: Large-scale simple question answering with memory networks. arXiv preprint arXiv:1506.02075 (2015)
6. Butt, S., Ashraf, N., Siddiqui, M.H.F., Sidorov, G., Gelbukh, A.: Transformer-based extractive social media question answering on tweetqa. Computación y Sistemas **25**(1) (2021)
7. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research **16**, 321–357 (2002)
8. De Cao, N., Izacard, G., Riedel, S., Petroni, F.: Autoregressive entity retrieval. arXiv preprint arXiv:2010.00904 (2020)
9. Dong, L., Lapata, M.: Language to logical form with neural attention. arXiv preprint arXiv:1601.01280 (2016)
10. Dubey, M., Banerjee, D., Abdelkawi, A., Lehmann, J.: Lc-quad 2.0: A large dataset for complex question answering over wikidata and dbpedia. In: International semantic web conference. pp. 69–78. Springer (2019)
11. Feng, S.Y., Gangal, V., Kang, D., Mitamura, T., Hovy, E.: Genaug: Data augmentation for finetuning text generators. arXiv preprint arXiv:2010.01794 (2020)
12. Ferragina, P., Scaiella, U.: Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In: Proceedings of the 19th ACM international conference on Information and knowledge management. pp. 1625–1628 (2010)
13. He, H., Bai, Y., Garcia, E.A., Li, S.: Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In: 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence). pp. 1322–1328. IEEE (2008)
14. Kobayashi, S.: Contextual augmentation: Data augmentation by words with paradigmatic relations. arXiv preprint arXiv:1805.06201 (2018)
15. Lan, Y., He, G., Jiang, J., Jiang, J., Zhao, W.X., Wen, J.R.: A survey on complex knowledge base question answering: Methods, challenges and solutions. arXiv preprint arXiv:2105.11644 (2021)
16. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.t., Rocktäschel, T., et al.: Retrieval-augmented generation for knowledge-intensive nlp tasks. arXiv preprint arXiv:2005.11401 (2020)
17. Mihindukulasooriya, N., Dubey, M., Gliozzo, A., Lehmann, J., Ngonga Ngomo, A.C., Usbeck, R., Rossiello, G., Kumar, U.: Semantic answer type and relation prediction task (smart 2021). arXiv (2022)
18. Mihindukulasooriya, N., Rossiello, G., Kapanipathi, P., Abdelaziz, I., Ravishankar, S., Yu, M., Gliozzo, A., Roukos, S., Gray, A.: Leveraging semantic parsing for relation linking over knowledge bases. In: International Semantic Web Conference. pp. 402–419. Springer (2020)
19. Ngomo, N.: 9th challenge on question answering over linked data (qald-9). language **7**(1), 58–64 (2018)

20. Nikas, C., Fafalios, P., Tzitzikas, Y.: Two-stage semantic answer type prediction for question answering using bert and class-specificity rewarding. In: SMART@ ISWC. pp. 19–28 (2020)
21. Perevalov, A., Both, A.: Augmentation-based answer type classification of the smart dataset. In: SMART@ ISWC. pp. 1–9 (2020)
22. Petroni, F., Piktus, A., Fan, A., Lewis, P., Yazdani, M., De Cao, N., Thorne, J., Jernite, Y., Karpukhin, V., Maillard, J., et al.: Kilt: a benchmark for knowledge intensive language tasks. arXiv preprint arXiv:2009.02252 (2020)
23. Piccinno, F., Ferragina, P.: From tagme to wat: a new entity annotator. In: Proceedings of the first international workshop on Entity recognition & disambiguation. pp. 55–62 (2014)
24. Reddy, S., Lapata, M., Steedman, M.: Large-scale semantic parsing without question-answer pairs. Transactions of the Association for Computational Linguistics **2**, 377–392 (2014)
25. Rossiello, G., Mihindukulasooriya, N., Abdelaziz, I., Bornea, M., Gliozzo, A., Naseem, T., Kapanipathi, P.: Generative relation linking for question answering over knowledge bases. arXiv preprint arXiv:2108.07337 (2021)
26. Sakor, A., Singh, K., Patel, A., Vidal, M.E.: Falcon 2.0: An entity and relation linking tool over wikidata. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management. pp. 3141–3148 (2020)
27. Setty, V., Balog, K.: Semantic answer type prediction using bert: Iai at the iswc smart task 2020. arXiv preprint arXiv:2109.06714 (2021)
28. Singh, J., McCann, B., Keskar, N.S., Xiong, C., Socher, R.: Xlda: Cross-lingual data augmentation for natural language inference and question answering. arXiv preprint arXiv:1905.11471 (2019)
29. Talmor, A., Berant, J.: The web as a knowledge-base for answering complex questions. arXiv preprint arXiv:1803.06643 (2018)
30. Trivedi, P., Maheshwari, G., Dubey, M., Lehmann, J.: Lc-quad: A corpus for complex question answering over knowledge graphs. In: International Semantic Web Conference. pp. 210–218. Springer (2017)
31. Wei, J., Zou, K.: Eda: Easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint arXiv:1901.11196 (2019)
32. Xu, K., Wu, L., Wang, Z., Yu, M., Chen, L., Sheinin, V.: Exploiting rich syntactic information for semantic parsing with graph-to-sequence model. arXiv preprint arXiv:1808.07624 (2018)
33. Yao, X., Van Durme, B.: Information extraction over structured data: Question answering with freebase. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 956–966 (2014)
34. Yih, W.t., Chang, M.W., He, X., Gao, J.: Semantic parsing via staged query graph generation: Question answering with knowledge base. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 1321–1331. Association for Computational Linguistics, Beijing, China (Jul 2015). https://doi.org/10.3115/v1/P15-1128, https://aclanthology.org/P15-1128
35. Yih, W.t., Richardson, M., Meek, C., Chang, M.W., Suh, J.: The value of semantic parse labeling for knowledge base question answering. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 201–206 (2016)