The Combination of BERT and Data Oversampling for Answer Type Prediction

Thang Ta Hoang[1,2][0000−0003−0321−5106], Olumide Ebenezer Ojo[1][0000−0003−3500−5218], Olaronke Oluwayemisi Adebanji[1][0000−0002−7412−6277], Hiram Calvo[1][0000−0003−2836−2102], 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, olumideoea@gmail.com, olaronke.oluwayemisi@gmail.com, hcalvo@cic.ipn.mx, gelbukh@cic.ipn.mx

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

Abstract. In this paper, we address the Task 1 (of the SMART Task 2021) of predicting the answer categories and types based on target ontologies, which could be useful in knowledge-based Question Answering (QA) systems. We introduced our method by combining the power of BERT architectures with data oversampling via replacements of linked terms to Wikidata and dependent noun phrases to attain the state-of-the-art performance. The accuracy on the DBpedia dataset is 98.5%, whereas NDCG@5 and NDCG@10 are 72.7% and 66.4% respectively. Our model has the best performance compared to other teams, with the accuracy score of 98% and Mean Reciprocal Rank (MRR) of 70% on the Wikidata dataset.

Keywords: Question Answering · Answer Type Prediction · Semantic Web Challenge · ISWC.

1 Introduction

In Natural Language Processing (NLP), Knowledge Base Question Answering (KBQA) is a task that involves searching for correct answers to a given natural language question using knowledge bases (KB). While this task appears easy to humans, it becomes a challenge for machines to detect the semantics in the questions and match them to the KBs before choosing which answer is the best. Language diversity, as well as semantic meanings, are barriers that create lexical gaps between questions and answers in reality. At the moment, there are two main approaches: semantic parsing-based (SP-based) methods and information retrieval-based (IR-based) methods, with deep neural networks making substantial contributions to improving their performance. [6, 12, 27].

In SMART 2021 Semantic Web Challenge[3], we engage Task 1 – category and type prediction over two datasets, Wikidata and DBpedia [13]. Given a

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

natural language question, our task is to build models that can predict answer category and answer type by relying on a set of candidates from a target ontology. There are 3 answer categories, "`resource`", "`literal`" and "`boolean`". If the answer type is "`boolean`", the answer category will also be "`boolean`". If the answer category is "`literal`", the answer type can be either "`number`", "`date`" or "`string`". If the answer category is "`resource`", the answer type will be an ontology class (DBpedia or Wikidata). Category prediction will be evaluated by accuracy scores across all datasets. A metric lenient NDCG@k [1] (NDCG@5 and NDCG@10) will be applied to the DBpedia dataset, and the Wikidata dataset will be evaluated with a mean reciprocal rank (MRR).

In this paper, we apply BERT classifiers [23] and use an oversampling method based on replacements of linked terms [5] and dependent noun phrases analyzed from the questions on the dataset. The paper is organized as follows: related works will be presented in Section 2, followed by data analysis in Section 3. In Section 4, we go over our methodology about augmenting and dividing the dataset for the training process. Experimental results and error reports with discussions are presented in Sections 5 and 6 respectively. Finally, in Section 7, we focus on certain conclusions and describe further work.

2 Related Works

As the task description of this challenge, each question should belong to a unique category class. As a result, category prediction is a multiclass classification problem. On the other hand, type prediction is a multilabel classification in which a given question can have multiple answer types. With answer categories and types predicted, a KBQA system can reduce the searching time for possible answers in the data [24].

If we consider category and type prediction as a problem of text classification, there is a wide range of methods from logistic regression (LG), Naive Bayes or Bayes networks [9] [17], support vector machine (SVM) [16], random forest (RF) [19], k-nearest neighbors (k-NN) [7], to deep networks (CNN, RNN, GCN) [11] [15] and many more hybrid approaches [6, 12, 27]. Due to the large number of types in the corpus, type prediction is sometimes considered a translation task in sequence-to-sequence models [20, 26], where a question is directly inferred to answer types/relations directly or during knowledge validation.

Referring to some of the papers from the previous years' challenge [14, 18, 21], BERT outperformed other approaches in predicting answer categories and types. As a result, we opt to use BERT to tackle the problem.

KBQA corpora are usually imbalanced with numerous natural language questions, including rare questions created from language diversity and human creativity. Therefore, oversampling and undersampling techniques are helpful to reduce popular data and increase rare data until the dataset distribution is more balanced or less biased. There are many methods known for oversampling techniques such as SMOTE [2], ADASYN [8], and data augmentation (EDA [25], GenAug [4], contextual augmentation [10]) for text. In this paper, we employ a

simple oversampling technique to limit the number of rare types with fewer than 5 questions. Our technique will replace dependent noun phrases with the roots and terms linked [5] to Wikidata by their aliases. Besides, cross-lingual data augmentation should be applied to improve the questions in different languages as well as overall outcome performance [22].

3 Dataset Analysis

To determine the proper approach, we will first do a preliminary analysis of the datasets. Both Wikidata and DBpedia datasets are unbalanced and are classified into three categories: `resource`, `literal` and `boolean`. The resource category accounts for the majority of datasets, followed by literal boolean categories. We conduct some investigations on older datasets, which are not significantly different from newer datasets with noise removed.

In the DBpedia dataset, we found a lot of questions with "empty" or `None` answer types, as shown in Figure 1. One of the solutions for this problem is that we can use a natural language inference (NLI) approach [3] to infer similar questions, we then can grant new types to these questions. However, we prefer to remove these empty types out of the training models and consider them as noises.

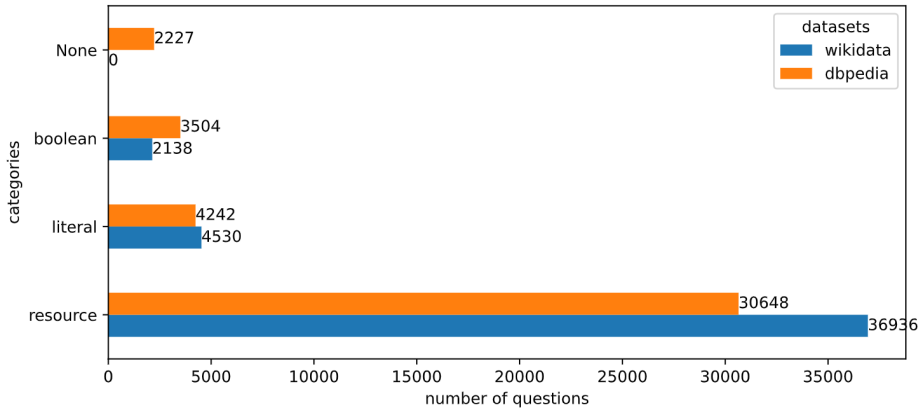

Fig. 1. The distribution of answer categories over 2 datasets: Wikidata and DBpedia. The `None` category occurs only in the DBpedia dataset.

Type prediction is not difficult with the `boolean` and `literal` categories because their answer types are simple. There are only a single type (`boolean`) for the `boolean` category and three types: (`date`, `number`, and `string`) for the `literal` category. In contrast, it is problematic with the `resource` category because it contains 376 types of DBpedia and 3308 types of Wikidata. To reduce

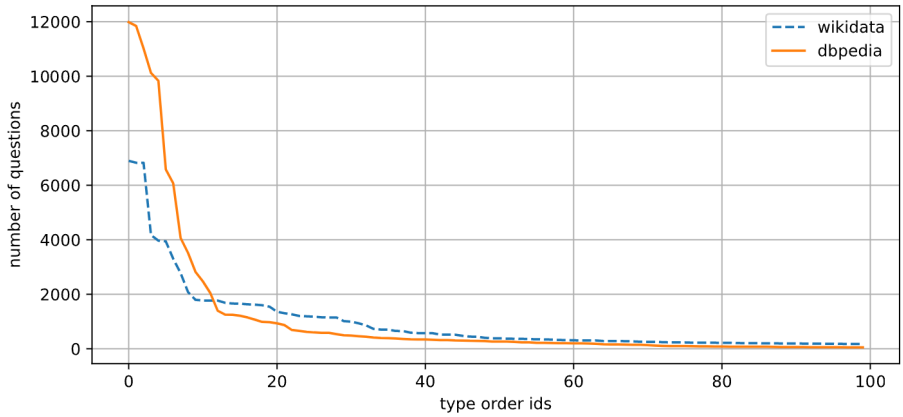

Fig. 2. The distribution of answer types over 2 datasets, Wikidata and DBpedia in 100 first ranked types by the number of questions.

the number of types in Wikidata, we only take the first type of answer that will be used for the experiment. We can also use ontology taxonomy of Wikidata to filter the redundant types (parent-child cases) to get the lesser number of types, but we will not do so in this work.

The imbalanced features in the datasets appear in the answer types as shown in Figure 2. When taking the first type, which is prepared for the training process, we notice that there are still a lot of types with only a few questions. There are 2032 types and 128 types corresponding to Wikidata and DBpedia datasets, respectively, if counting response types with less than 5 questions. Clearly, it is a challenge for text classification methods to detect enough differences between rare types. We debated whether to remove these rare types or keep them because they do not contribute so much to the classification performance. Finally, we decided to keep these types and apply an oversampling technique to increase the number of questions of each type as many as possible, with the expectation that the datasets are less imbalanced than before.

4 Our Methodology

4.1 Preprocessing and Oversampling Methods

For each question, spaCy[4] v.2.3.2 was used to analyze the question structure to get its semantic units, such as question type, subject, main verb (also `ROOT`), terms (noun phrases, dependent noun phrases) to build the sentence template, and apply entity linking (EL) methods to extract terms connected to Wikidata. For spaCy components, we take `en_core_web_lg`, `STOP_WORDS`, `lemmatizer` (`Lemmatizer`, `ADJ`, `NOUN`, `VERB`), and `sentencizer` pipeline.

[4] https://spacy.io/

The sentence template was built by a greedy algorithm which absorbs all longest terms. First, we rank the terms according to their length. Then, for each term, we search it in the sentence. If found, we put a pair of opening and closing curly brackets {...} around it to form the sentence template. This algorithm will ignore terms that are contained within larger ones.

We realized that the question structure is also useful for Task 2 of this challenge, which involves predicting the relations of questions. To easier understand what we analyzed, take a look at the following example.


```
{
"question": "What periodical literature does Delta Air Lines use
↪  as a moutpiece?",
"category": "resource",
"type": ["publication", "recurring", "intellectual work", "text",
↪  "communication medium", "serial"],
"question_template": "What {periodical literature} does {Delta Air
↪  Lines} {use} as {a mouthpiece}?",
"key_terms": ["periodical literature", "Delta Air Lines", "use",
↪  "a mouthpiece"],
"subject": ["Delta Air Lines"],
"main verb": ["use"],
"entities": ["literature", "a mouthpiece", "mouthpiece"],
"question_type": "what",
"dependent_nouns": ["Delta Air Lines", "periodical
↪  literature",...],
"el_terms": [ "Delta Air Lines": { "wikidata_id": "Q188920",
↪  "label": "Delta Air Lines", "aliases": ["DAL", "Delta Air
↪  Lines, Inc."},...],
...
}
```


There are some cases where terms are typos, so grammatical models should be used to correct them. For each term, we depend on the API searching[5] of Wikipedia to fix the typos for terms longer than eight characters. We assume that a short term can lead to a wrong search. In some cases, the result may be a term that is more popular than the term we desire. As a result, we may have a wrong fix that distorts the meaning of the original sentence. In the example, we apply this API to fix the typo `moutpiece` to `mouthpiece`.

For the EL methods, we already built APIs for performing ELs using various methods, such as Babelfy, OpenTapioca, Wikifier, and AIDA but we decided to use TagMe, with WAT as an alternative. We gather linked terms to Wikidata which have the link probability higher than 0.9. After that, we apply an oversampling technique over linked terms to increase the number and discrepancy of questions over rare types. We simply replace aliases of linked terms to

[5] https://www.mediawiki.org/wiki/API:Search

their labels to create new questions. In the example, we can produce some new questions like:

```
{
"question": "What periodical literature does Delta Air Lines use
↪  as a moutpiece?",
"new questions": ["What periodical literature does DAL use as a
↪  mouthpiece?", "What periodical literature does Delta  Air
↪  Lines,  Inc. use as a mouthpiece?","What literature does Delta
↪  Air Lines use as a moutpiece?",...],
...
}
```

Table 1. The comparisons between original and extended datasets. There are 376 types in DBpedia datasets and 3308 types in Wikidata datasets. The number of rare types is counted for the answer types with fewer than five questions.

	Original datasets		Extended datasets	
	Questions	**Rare types**	**Questions**	**Rare types**
DBpedia	40621	128	126872	77
Wikidata	43604	2032	150077	1180

We generated more questions by replacing key terms to their roots, such as `periodical literature` to `literature`. We assume that by using these replacements, the models will be able to deal with the new data more effectively.

By using our strategy, we were able to adjust the number of questions and rare types between the original and extended datasets, as shown in Table 1. For each question, we only count single types rather than flattening type lists into strings. In comparison to the original dataset, the new datasets have nearly three times more and less in roughly half of rare types than the original dataset. The number of questions and answer types changes slightly when we apply some minor preprocessing measures before the training process.

4.2 Data Training and Models

According to our initial experiments, we decided to split our datasets into two groups (`D1` and `D2`) each of which will be trained by two BERT models, one for category prediction (also type prediction for the `literal` and `boolean` categories) and the other for type prediction of the `resource` category. In total, we have four BERT models for two datasets (two for DBpedia and two for Wikidata).

- `D1`: These datasets are used for category prediction and type prediction for the `boolean` and `literal` categories. They are flattened into 5 categories:

`boolean`, `literal-date`, `literal-number`, `literal-string`, and `resource` as seen in the work of Setty and co-authors [21]. We only use the `boolean` and `literal` categories in the extended data, not the `resource` category, because it already takes the majority of the original datasets.
– `D2`: These datasets contains the questions belonging to the `resource` category as well as their new questions generated from our oversampling method. We will not present the graph distribution of answer types for questions, in which the number of questions drops sharply from the popular to rare types.

We adapt BERT pre-trained models to a specific task, in this case text classification. We set dropout with `p = 0.1` and use linear regression to fit `hidden_size` to `n_classes`. In training process, we also use cross-entropy loss and Adam optimization with `learning rate = 2e-5`. Other hyperparameters are `MAX_LEN = 192`, `RANDOM_SEED = 42`, and `BATCH_SIZE = 8`. We use a small value of `BATCH_SIZE` due to the limited resources on our server.

Table 2. The training data of `D1` datasets.

Dataset	Boolean	Literal date	Literal number	Literal string	Resource	Total
DBpedia	11274	12957	5207	5035	30648	65121
Wikidata	10207	14391	4124	7548	36936	73206

Table 3. The training data of `D2` datasets. Rare types are counted for types having less than 5 samples.

Dataset	Questions	Types	Rare types	Type flatten
DBpedia	76550	1224	588	Yes
Wikidata	113807	1860	832	No

Table 2 shows the total number of types and rare types in the DBpedia and Wikidata datasets. We take all types of the DBpedia dataset and flatten type lists as strings for the training process because the metric NDCG@k was used. For example, list `["publication","recurring",...,"serial"]` will be converted to string `"publication,recurring,..., serial"`. In contrast, the Wikidata dataset is ranked by an MRR score, so we only take the first type (single type) of each question. A type that is in a higher position in the lists is assumed to contribute to a higher overall MRR score.

By comparing Table 1, the numbers of types of DBpedia datasets has changed as a result of type flattening and somewhat has a data correlation with those of Wikidata datasets in Table 2. This offers a fair enough way to evaluate models

across both DBpedia and Wikidata datasets if the same metric is used. Meanwhile, the numbers of types of Wikidata datasets are now slightly reduced via the oversampling step.

5 Experiments

5.1 Category and Type Prediction of Boolean and Literal Categories

We apply `bert-based-cased`[6] to `D1` datasets to predict the categories and types (only for the `literal` and `boolean` categories) over questions. As previously stated, the questions are divided into five categories: `boolean`, `literal-date`, `literal-number`, `literal-string`, and `resource`. By this way, the model can predict not only the answer category but also the answer type for questions which belong to the `literal` category and the `boolean` category. The only thing left to accomplish is to predict the answer types of questions in the `resource` category. This task will be completed in the following section.

We split `D1` datasets into three subsets: train test, test set, and validation set, in the ratio 8:1:1. After 10 epochs, we will save the best model based on the highest validation accuracy. Since the organizers do not initially provide us with the golden label set, we decide to use validation sets extracted from the original datasets.

Table 4. The training results of `D1` datasets on category prediction.

Dataset	Train acc	Test acc	Val acc
DBpedia	0.999	0.998	0.993
Wikidata	0.999	0.997	0.990

Table 4 shows that the accuracy scores of the Wikidata and DBpedia datasets are very high, as declared in previous works of the last year challenge. Our findings confirm that BERT models perform exceptionally well in predicting answer categories in KBQA systems.

5.2 Type prediction for the resource category

The previous section discusses type prediction of answers over the `literal` and `boolean` categories. In this section, we do the same thing, but for the type prediction of the `resource` category. We also apply `bert-based-cased` over `D2` datasets for answer types. As with `D1`, we divided `D2` into three subsets: training, testing and validation set with the ratio 8:1:1. Even though we apply an oversampling method to the original dataset, as described in Section 3.2, there are many rare types with fewer than five questions in the datasets.

[6] https://huggingface.co/bert-base-cased

Table 5. The training results of `D2` datasets on the type prediction over the resource category.

Dataset	Train acc	Test acc	Val acc
DBpedia	0.995	0.981	0.949
Wikidata	0.991	0.985	0.950

Table 5 shows the evaluation metrics on `D2` datasets after at least 15 epochs. Our suggestion is to train until `train_acc` is at the highest value as possible to maximize the model's ability to learn the rare types.

5.3 Compare with the organizers' results

Even though we achieve optimistic results from training data steps, we must work on the test sets offered by the organizers to have the final evaluations. We consider these sets as holdout sets, providing samples that do not appear mostly in the training data. The category prediction will use accuracy scores as the metric to evaluate our models. The metric lenient NDCG@k (NDCG@5 and NDCG@10) will be applied to evaluate the performance of the type prediction on the DBpedia dataset, whereas the MRR score will be used for the Wikidata dataset.

Table 6. The final evaluation metrics on the DBpedia dataset by teams.

Team	Accuracy	NDCG@5	NDCG@10
Chaeyoon	0.984	0.842	0.854
Bhargav	0.985	0.825	0.790
Remzi	0.985	0.725	0.704
Our team	**0.985**	**0.727**	**0.664**
Nadine	0.991	0.734	0.658
Aleksandr	0.991	0.643	0.577

Table 7. The final evaluation metrics on the Wikidata dataset by teams.

Team	Accuracy	MRR
Our team	**0.98**	**0.70**
Remzi	0.98	0.66
Nadine	0.99	0.45
Aleksandr	0.98	0.43

In Table 6, we obtained the accuracy score of 98.5%, NDCG@5 of 0.727, and NDCG@10 of 0.664 for the DBpedia dataset. When compared to the best performance, our results are among the best and quite competitive. Meanwhile, our method performs best on Wikidata, with an accuracy of 98% and MRR of 0.7 as shown in Table 7.

6 Error Reports and Discussions

When working with the old datasets, we detect a tiny number of errors relating to `null` answer types or the improper agreement between questions and answer types. Some questions are too short and can not be analyzed precisely to get semantic units, while others are too long. We consider them all as noises and filter them out before training the data.

Our analysis of question structure is not always correct. In some cases, we are unable to obtain the necessary information, such as the question type, subject, main verb, etc. The parsing method occasionally detects wrong dependent nouns as shown in Table 8. In the first example, the term `'s husband` leads to generate a new wrong question "`Who is the child of Ranavalona Ihusband?`" when replacing `'s husband` by `husband` (considered as the noun root). As a result, we must avoid all replacements on possessive nouns containing `'s`. The second example has no effect on the replacements, but the parsing cannot split the long term `right ascension of malin 1` into two smaller terms, `right ascension` and `malin 1` to produce new questions.

Table 8. Two examples with errors in detecting wrong key terms.

Example 1	
question	Who is the child of Ranavalona I's husband?
key_terms	"the child", "Ranavalona I", " 's husband "
question_template	Who is {the child} of {Ranavalona I}{ 's husband }?
Example 2	
question	Is the right ascension of malin 1 less than 15.1398?
key_terms	" right ascension of malin 1 ", "15.1398"
question_template	Is the { right ascension of malin 1 } less than {15.1398} ?

Replacing dependent noun phrases in the creation of new questions does improve the accuracy of category prediction. However, our intuition tells us that we can distort the meaning of questions even in the least aspect. As a result, this may affect type prediction. We thus avoid applying this task to any dependent noun phrases that are fewer than 8 characters long. We assume that the longer phrases can keep the original meaning better. Since we only take linked terms to Wikidata with those that have a link probability score over 0.9, we may leave a lot of other helpful but equally correct linked terms with lower probabilities.

Due to the occurrence of so many rare types, there are likely not enough rare types in the training set after splitting the data into training, testing, and validation sets. On one side, it helps the model's detection of similar types when the rare types do not appear. On the other hand, this splitting may have an impact on the performance of predicting rare types. Hence, we should place all data on the training set or retrain the model until the accuracy score achieves the best convergence across all sets.

In the DBpedia dataset, the performances are relatively low in comparison to other teams. This happens when we flatten answer type lists to strings, instead of having to use different models to detect single answer types based on their ranking.

7 Conclusion

In this paper, we participated in Task 1 of the SMART 2021 Semantic Web Challenge, category and type prediction of answers using a set of hint ontologies. We apply spaCy and TagMe to extract sentence components and linked terms from questions. By using a simple oversampling method based on replacements of linked terms and dependent nouns, we are able to expand the size of datasets about three times, targeting to have as many questions as possible, especially on rare answer types. Our project code can be accessed at GitHub [7].

In the experiments, a pretrained BERT model, `bert-base-cased`, was used to train `D1` and `D2` datasets to predict answer categories and types. For DBpedia, NDCG@5 and NDCG@10 are 0.727 and 0.664 respectively, with an accuracy of 98.5%. The best results in the Wikidata dataset have an MRR score of 0.7 and an accuracy of 98%. We discover that BERT models perform well in multiclass and multilabel classification problems.

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 plan to test various neural networks or hybrid approaches to search for a superior method, as well as try to augment the dataset using entity linking methods and multilingual translation. Finally, the semantic relationships between answer types should be studied by linking to questions in order to minimize the number of types and infer the answer types effectively.

Acknowledgement

The work was done with partial support from the Mexican Government through the grant A1-S-47854 of the CONACYT, Mexico, and by the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico, under Grants 20211884, 20220859, and 20220553, EDI; and COFAA-IPN. The authors thank the CONACYT for the computing resources brought to them through the

[7] https://github.com/thangth1102/SMART_2021_Task1

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. Balog, K., Neumayer, R.: Hierarchical target type identification for entity-oriented queries. In: Proceedings of the 21st ACM international conference on Information and knowledge management. pp. 2391–2394 (2012)
2. 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)
3. Demszky, D., Guu, K., Liang, P.: Transforming question answering datasets into natural language inference datasets. arXiv preprint arXiv:1809.02922 (2018)
4. Feng, S.Y., Gangal, V., Kang, D., Mitamura, T., Hovy, E.: Genaug: Data augmentation for finetuning text generators. arXiv preprint arXiv:2010.01794 (2020)
5. 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)
6. Fu, B., Qiu, Y., Tang, C., Li, Y., Yu, H., Sun, J.: A survey on complex question answering over knowledge base: Recent advances and challenges. arXiv preprint arXiv:2007.13069 (2020)
7. Han, E.H.S., Karypis, G., Kumar, V.: Text categorization using weight adjusted k-nearest neighbor classification. In: Pacific-asia conference on knowledge discovery and data mining. pp. 53–65. Springer (2001)
8. 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)
9. Kafle, K., Kanan, C.: Answer-type prediction for visual question answering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4976–4984 (2016)
10. Kobayashi, S.: Contextual augmentation: Data augmentation by words with paradigmatic relations. arXiv preprint arXiv:1805.06201 (2018)
11. Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., Brown, D.: Text classification algorithms: A survey. Information **10**(4), 150 (2019)
12. 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)
13. 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)
14. 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)
15. Ojo, O., Ta, T., Adebanji, O., Gelbukh, A., Calvo, H., Sidorov, G.: Automatic hate speech detection using deep neural networks and word embedding. Computación y Sistemas **26** (2022)

16. Ojo, O.E., Gelbukh, A., Calvo, H., Adebanji, O.: Performance study of n-grams in the analysis of sentiments. Journal of the Nigerian Society of Physical Sciences pp. 140–143 (2021)
17. Ojo, O.E., Gelbukh, A., Calvo, H., Sidorov, G., Adebanji, O.: Sentiment analysis in texts on economic domain. In: Proceedings of the 19th Mexican International Conference on Artificial Intelligence - MICAI2020. Mexico City, Mexico (October 2020)
18. Perevalov, A., Both, A.: Augmentation-based answer type classification of the smart dataset. In: SMART@ ISWC. pp. 1–9 (2020)
19. Pranckevičius, T., Marcinkevičius, V.: Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification. Baltic Journal of Modern Computing **5**(2), 221 (2017)
20. Rossiello, G., Mihindukulasooriya, N., Abdelaziz, I., Bornea, M., Gliozzo, A., Naseem, T., Kapanipathi, P.: Generative relation linking for question answering over knowledge bases. In: International Semantic Web Conference. pp. 321–337. Springer (2021)
21. Setty, V., Balog, K.: Semantic answer type prediction using bert: Iai at the iswc smart task 2020. arXiv preprint arXiv:2109.06714 (2021)
22. 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)
23. Sun, C., Qiu, X., Xu, Y., Huang, X.: How to fine-tune bert for text classification? In: China National Conference on Chinese Computational Linguistics. pp. 194–206. Springer (2019)
24. Wasim, M., Asim, M.N., Khan, M.U.G., Mahmood, W.: Multi-label biomedical question classification for lexical answer type prediction. Journal of biomedical informatics **93**, 103143 (2019)
25. Wei, J., Zou, K.: Eda: Easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint arXiv:1901.11196 (2019)
26. Wu, L., Wu, P., Zhang, X.: A seq2seq-based approach to question answering over knowledge bases. In: Joint International Semantic Technology Conference. pp. 170–181. Springer (2019)
27. Wu, P., Zhang, X., Feng, Z.: A survey of question answering over knowledge base. In: China Conference on Knowledge Graph and Semantic Computing. pp. 86–97. Springer (2019)