Augmentation-based Answer Type Classification of the SMART dataset

Aleksandr Perevalov and Andreas Both

Anhalt University of Applied Sciences, Köthen (Anhalt), Germany
{aleksandr.perevalov,andreas.both}@hs-anhalt.de

Abstract. Recent progress in deep-learning-enabled AI researchers and developers to invest minimal efforts to achieve state-of-the-art results. Specifically, in such a task as text classification – text preprocessing and feature generation does not play a significant role anymore thanks to such a landmark model as BERT and other related models. In this paper, we present our solution for the Semantic Answer Type prediction task (SMART task). The solution is based on the application of several data augmentation techniques: machine translation to popular languages, round-trip translation, named entities annotation with linked data. The final submission was generated as a weighted result from several successful system outputs.

Keywords: Answer type classification · Text classification · Text augmentation.

1 Introduction

Understanding a question's answer type is one of the significant steps in a question-answering process [4]. With the help of an answer type classifier – a Question Answering system (QA system) could narrow the answer search space and filter the inappropriate answer candidates [6].

In general, the answer type classification task can be interpreted as a multi-class text classification task. However, the SMART task [5] proposes a more complicated structure of the data. There are two class levels: answer category (resource, literal, boolean) and answer type.

According to the official description of the data[1]: If the category is "resource", answer types are ontology classes from either the DBpedia ontology[2] or the Wikidata ontology[3]. If the category is "literal", answer types are either "number", "date", or "string". For the category "boolean" no additional specialization is defined. It is worth mentioning that in this work we concentrate only on the DBpedia dataset.

---

[1] https://smart-task.github.io/

[2] http://mappings.dbpedia.org/server/ontology/classes/

[3] https://www.wikidata.org/wiki/Wikidata:WikiProject_Ontology

Each "resource" answer type contains a ranked list of the DBpedia ontology types. All items contained in a list are part of one hierarchy, for example: `["dbo:Person", "dbo:Agent"]` or `["dbo:Opera", "dbo:MusicalWork", "dbo:Work"]`. The most general ontology type is at the end of a list.

The DBpedia dataset contains 21,964 (train - 17,571, test - 4,393) questions. The evaluation metric for answer category prediction task is **accuracy**, the metric for answer type prediction is **lenient NDCG@k with a Linear decay** [2].

Our solution focuses on data augmentation techniques. In Section 2 we describe the dataset in detail. Section 3 incorporates the description for the data augmentation methods used by us, as well as an algorithm for merging answer type lists. In Section 4 we show our experimental results and describe the local evaluation pipeline. Finally, in Section 5 the conclusions are presented.

2   Dataset analysis and transformation

The original dataset is presented using the JSON format. To train a model on the data, it needs to be transformed into a feature-target form.

In the case of the prediction answer category, the task is trivial – there is just one target value for one question and it is considered as a multi-class classification task. While predicting an answer type – things are more complicated: we have to predict a list, which items are ordered according to the level of taxonomy and has to match one hierarchy (e.g., dbo:Opera, dbo:MusicalWork, and dbo:Work). The first constraint does not allow us to consider this task as a multi-class classification. That is why we decided to make each item of a list as an individual target value, so we can train separate models for each of them. We took only 5 most general types for each question because 95% of the answer type list's lengths are not more than this value. The head of the resulting dataset is presented in Figure 1.

	id	question	category	type_1	type_2	type_3	type_4	type_5
0	dbpedia_1177	Was Jacqueline Kennedy Onassis a follower of M...	boolean	NaN	NaN	NaN	NaN	NaN
1	dbpedia_14427	What is the name of the opera based on Twelfth...	resource	dbo:Work	dbo:MusicalWork	dbo:Opera	NaN	NaN
2	dbpedia_16615	When did Lena Horne receive the Grammy Award f...	literal	date	NaN	NaN	NaN	NaN
3	dbpedia_23480	Do Prince Harry and Prince William have the sa...	boolean	NaN	NaN	NaN	NaN	NaN
4	dbpedia_3681	What is the subsidiary company working for Leo...	resource	dbo:Agent	dbo:Organisation	dbo:EducationalInstitution	NaN	NaN
5	dbpedia_3712	what is the musical composer id of bedrish sme...	literal	string	NaN	NaN	NaN	NaN
6	dbpedia_14847	How often are the Paralympic games held?	literal	number	NaN	NaN	NaN	NaN

Fig. 1. Tabular representation of the training dataset.

Hence, we consider the solution for the SMART challenge task to be represented as two-level architecture where the higher-level decisions activate lower-level classifiers:
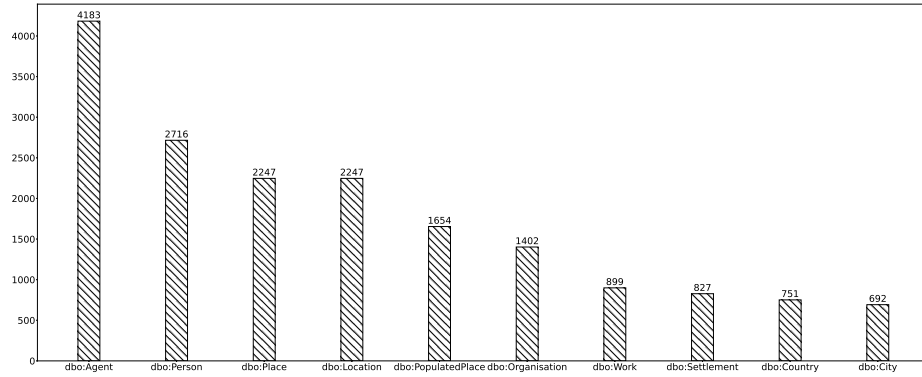
Fig. 2. TOP 10 resource answer types

Level 1 The category is classified (Figure 1, column: "category"). Thereafter, the classification system can decide for the next required classifiers.

Level 2 The second-level decisions are considered to be two independent tasks:
- Classification of literal type (Figure 1, column: "type_1")
- Classification of resource types (Figure 1, columns: "type_1", "type_2", "type_3", "type_4", "type_5")

The training dataset had 43 questions with empty textual representation. These questions were removed. The resulting dataset has the following characteristics:

- 17,528 questions are contained;
- Distribution: 9,573 question point to resources, 5,156 point to a literal datatype and 2,799 are Boolean questions;
- The 95th percentile of the answer type lists' length is 5;
- The maximum number of tokens in a question is 60.

In Figure 2, the top 10 most common resource answer types are presented. It shows that all top 10 resource types belonging either to `dbo:Agent` or `dbo:Place` or their sub-classes.

3   Proposed solution

3.1   Classifier Architecture

The classification pipeline was created with a tree-like structure and 7 classifiers in total (see Figure 3). First, the category is classified. Then, depending on the category, the corresponding models are chosen.

For example, if the category is "resource", then the pipeline classifies a question using 5 models reflecting the decision for "type_1", "type_2", "type_3",

"type_4", and "type_5" (cf., Figure 1). Given the results of these classifiers, the answer type list is created from the computed results (obeying the correct order). As there are only 5 models (one model for one list item) – the answer type list's length will contain no more than 5. Sometimes it may be less (when the prediction is `None`).
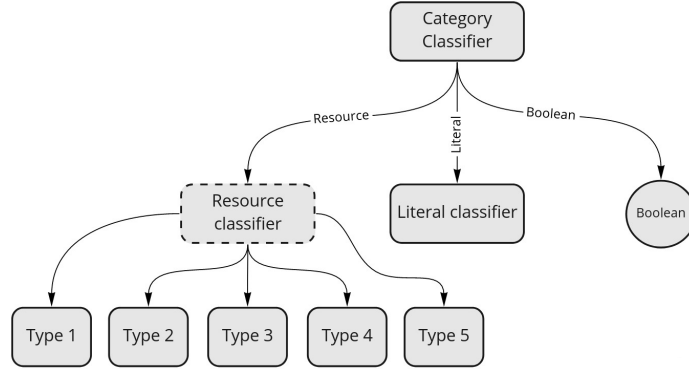

Fig. 3. Tree-like classification pipeline C

3.2   Data Augmentation

To extend the training data, we used several augmentation strategies for the given dataset:

D_1 Machine translation to German, French, Spanish, and Russian is used for each question. Hence, in total there are 5x more questions (separated in 5 different languages) resulting in 87,640 questions. As the dataset has become a multilingual one, we will use a multilingual model. There are two types of prediction for such a dataset: Use the original English text or use predictions for all languages and a majority voting algorithm.

D_2 Round-trip translation [1] (English-German-English, English-Russian-English) – in total, there are 3x more questions, and we use a single language model. The dataset consists of 52,584 questions;

D_3 Each question is annotated with it's named entities pointing to DBpedia resources – each named entity is replaced with one of its RDF types. The data is extracted from DBpedia with help of DBpedia Spotlight[4]. The dataset consists of 163,488 questions.

Google Cloud Translation[5] was used to translate the data for D_1 and D_2 automatically.
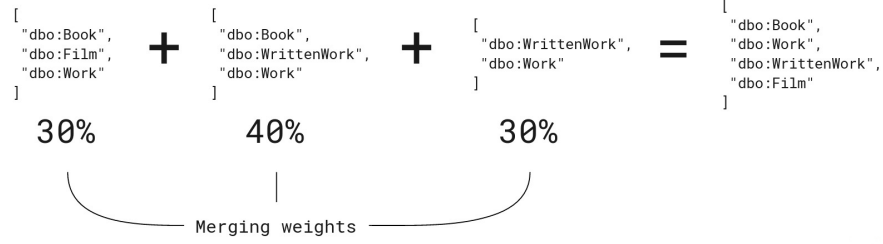
---

[4] https://www.dbpedia-spotlight.org/
[5] https://cloud.google.com/translate

Fig. 4. Example of merging 3 lists with specified weights

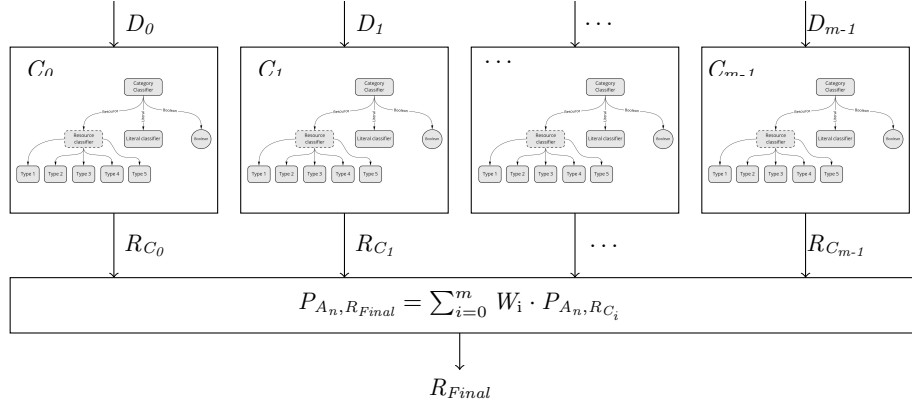$$P_{A_n, R_{Final}} = \sum_{i=0}^{m} W_i \cdot P_{A_n, R_{C_i}}$$

Fig. 5. Overview of the final process.

Hence, additionally to the original dataset – we call it D_0 – we have created here 3 more dataset (D_1, D_2, and D_3) that are used to spawn 4 independent classifier pipelines (C_1, C_2, C_3, and C_4). Consequently, the results R_{C_i} of all classifier pipelines C_i need to be merged. Figure 4 shows an example of merging process. The next section gives a detailed description of the process.

3.3   Results Merging

Each classification pipeline – C_0, C_1, C_3, and C_4 – provide a list of classification results. It is reasonable to assume that they also have a distinguished classification quality.

Hence, while merge the classification results – identified by R_{C_0}, R_{C_1}, R_{C_2}, and R_{C_3} – to establish a final result set R_{Final} as shown in Figure 5. The merging of R_{C_i} with $i \in \{0, 1, 2, 3\}$ is computed while numerically calculating a weighted rank for each answer type that was predicted by at least one classifier pipeline

C_i. The rank $P_{A_n, R_{Final}}$ of an answer type A_n in R_{Final} is computed as follows:

$$P_{A_n, R_{Final}} = \sum_{i=0}^{m} W_i \cdot P_{A_n, R_{C_i}}, \text{ where } P_{A_n, R_{C_i}} = \begin{cases} \text{rank of } n \text{ in } R_{C_i} & \text{if } n \text{ in } R_{C_i} \\ \text{fallback rank } f & \text{else} \end{cases}$$

$$\text{and } m \text{ is the number of classification pipelines}$$

Typically, the quality of Level-1 decisions would be high. However, there also exists a special case where a different answer category was predicted by the classifier pipelines. In this case, we currently follow a static rule-based decision process that is favoring the more specific predictions, i.e., if one classifier pipeline predicted the category `boolean`, then all other results are discarded. And, else if one classifier pipeline is predicting the `literal` category, then all non-`literal` categories are discarded.

4   Experiments

4.1   Evaluation

We used Bert-base-cased and Bert-base-multilingual-cased models [3] in our classification pipeline. Training data was split into two sets: train and validation set. The validation set was created by random choice of 4400 questions and the test set consists of 4381 questions. The models were fine-tuned on the training set with the following hyperparameters: `EPOCHS`=2, `MAX_LEN`=60, `BATCH_SIZE`=16.

The training process was performed on GPU resources provided by the Kaggle.com platform (NVIDIA TESLA P100 GPU, 16 GB RAM). The results shown in Table 1 enable us to compare the effectiveness of each augmentation technique. The results were obtained on the validation set locally (MV – corresponds to Majority Voting algorithm, see Section 3.2):

Table 1. Local validation results

	D_0	D_1	D_{1+MV}	D_2	D_3
Accuracy	0.969	0.968	0.962	0.357	0.959
NDCG@5	0.533	0.704	0.708	0.165	0.363
NDCG@10	0.499	0.661	0.665	0.140	0.317

The best performing datasets are multilingual ones (D_1). The round-trip translation (D_2) approach caused overfitting because of small differences in questions forms. The same situation occurred with the named entities annotation approach (D_3). The original dataset (D_0) showed comparable performance. A detailed analysis of the errors is given in Section 4.2.

For the final analysis, we took only predictions from the models trained on the original (D_0) and the multilingual dataset (D_1) into account. We used both prediction approaches for the multilingual data: using the multilingual model to

predict the answer type of English questions and using the same model while retrieving predictions for all 5 languages and taking the majority vote result. The predictions were merged using the algorithm described at the end of the previous section, we used several weights combinations to achieve the highest quality. The evaluation results for final submission are presented in Table 2.

Table 2. Final evaluation results

	$.3D_0+.3D_1+.4D_{1+MV}$	$.5D_1+.5D_{1+MV}$	$.3D_1+.7D_{1+MV}$	$.7D_1+.3D_{1+MV}$
Accuracy	0.976	0.965	0.965	0.972
NDCG@5	0.762	0.752	0.752	0.759
NDCG@10	0.725	0.714	0.716	0.722

The highest score on the test dataset was achieved with a merged combination of 3 predictions (see the second column of Table 2). We evaluated the weight combinations where each weight w_i was chosen between 0.0 and 1.0, s.t., the sum of all used weights equals 1.0. The following best weight combination was created using this process: $30\% - D_0$, $30\% - D_1$ and $40\% - D_{1+MV}$. The fallback rank f for the merging algorithm was taken equal to 10 (see Subsection 3.3). This combination was submitted as the final solution for the task. As the weights were obtained manually and intuitively, we can not make a statement about its application on the other datasets. Moreover, these weights can be overfitted to the test set because the final predictions were given by the organizers based on the whole test dataset without private/public test splits. Hence, the weights were selected according to the public test set results. This is a limitation of our merging approach.

4.2   Error analysis

As we reported in the previous subsection, the approach D_1 outperformed D_2 and D_3 due to the model overfitting caused by the nearly same surface form of the obtained questions. The corresponding example of D_2 is given below:

Original: Who replaced Charles Evans Hughes as the Chief Justice of The United States?

En-De-En: Who succeeded Charles Evans Hughes as Chief Justice of the United States?

En-Ru-En: Who replaced Charles Evans Hughes as Chief Justice of the United States?

Hence, we have to recognize that the questions generated using round trip translation are not differing significantly: En-De-En differs in one word, absence of the definite article, and non-capitalized letter "T" in the last definite article, almost the same is true for the En-Ru-En translation.

We can assume that round-trip translation to languages, that are non-popular or distant from the English language, would possibly resolve this issue.

The example of D_3 is given below:

Original: Who replaced Charles Evans Hughes as the Chief Justice of The United States?

Variant 1:   Who replaced `DBpedia:Athlete` as the `DBpedia:Person` of The `DBpedia:PopulatedPlace`?

Variant 2:   Who replaced `DBpedia:Person` as the `DBpedia:Person` of The `DBpedia:Country`?

Each named entity was replaced with its URI's type in the DBpedia. As a resource in the DBpedia may contain up to several thousands of variants corresponding to each combination of the types. There are two major limitations of this approach: the DBpedia resource may contain errors w.r.t. its type and the Named Entity Linking tool may extract and link entities incorrectly. In the given example, the "the Chief Justice of The United States" should be replaced with a single type, while it was replaced with two different types which are incorrect.

The D_3 showed the best performance, here is the example of its fragment:

Original: Who replaced Charles Evans Hughes as the Chief Justice of The United States?

German: Wer hat Charles Evans Hughes als Oberster Richter der Vereinigten Staaten abgelöst?

French: Qui a remplacé Charles Evans Hughes en tant que juge en chef des États-Unis?

However, despite the augmentation approaches, there is one significant limitation of our prediction approach – each element of the answer type list is predicted independently and therefore the elements may not from the same hierarchy. For example, for the question "What is the horse characters of Madame Sans-Gêne play?" predicted answer type list is `["dbo:Person", "dbo:Work"]` while the true value is `["dbo:Animal", "dbo:Eukaryote", "dbo:Species"]`. Despite the prediction is completely incorrect, it has items `"dbo:Person"` and `"dbo:Work"` which are located in the different ontology branches (hierarchies).

Consequently, the mechanism of checking the correctness of the hierarchy should be created. One of the possible solutions may be the prediction of the most specific answer type and making the prediction according to the actual hierarchy.

5   Conclusion

In this work, we described our solution for the Semantic Answer Type prediction task. The goal was to predict the corresponding answer category and answer types. To solve the task, we created a tree-like classification pipeline and implemented several text augmentation methods described in Section 3.

The results of our experiments show that the multilingual dataset has the highest performance in contrast to the other augmented data. To prepare the

final submission, we used the weighted merging algorithm on top of our best predictions (see Section 4).

Obviously, there is room for improvement. In future work, we would use an ensemble learning approach to merge the results instead of the current static approach. Additionally, we would also consider each language classifier independently assuming a distinguished translation quality leading to different classification quality. Also, the hierarchy accordance and hierarchy level validation mechanism might be used for the prediction process.

References

1. Aiken, M., Park, M.: The efficacy of round-trip translation for mt evaluation. Translation Journal **14**(1), 1–10 (2010)
2. 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). https://doi.org/10.1145/2396761.2398648
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.N.: Bert: Pre-training of deep bidirectional transformers for language understanding. ArXiv e-prints (2018)
4. Hao, T., Xie, W., Wu, Q., Weng, H., Qu, Y.: Leveraging question target word features through semantic relation expansion for answer type classification. Knowledge-Based Systems **133**, 43 – 52 (2017). https://doi.org/https://doi.org/10.1016/j.knosys.2017.06.030
5. Mihindukulasooriya, N., Dubey, M., Gliozzo, A., Lehmann, J., Ngomo, A.C.N., Usbeck, R.: SeMantic AnsweR Type prediction task (SMART) at ISWC 2020 Semantic Web Challenge. CoRR/arXiv **abs/2012.00555** (2020), https://arxiv.org/abs/2012.00555
6. Xu, D., Jansen, P., Martin, J., Xie, Z., Yadav, V., Madabushi, H.T., Tafjord, O., Clark, P.: Multi-class hierarchical question classification for multiple choice science exams. In: Proceedings of The 12th Language Resources and Evaluation Conference. pp. 5370–5382 (2020)