Similarity Detection of Natural-Language Questions and Answers using the VANiLLa dataset

A Both[1, 2], A Gashkov[3], M Eltsova[4]

[1] Professor, Department for Computer Science and Languages, Anhalt University of Applied Sciences, Lohmannstraße 23, 06366 Köthen, Germany
[2] Head of Research, DATEV eG, Nuremberg, Germany
[3] Master Student, Anhalt University of Applied Sciences, Lohmannstraße 23, 06366 Köthen, Germany
[4] Assistant Professor, Department for Foreign Languages and Public Relations, Perm National Research Polytechnic University, 13 Professora Pozdeeva st., 614013 Perm, Russia

E-mail: andreas.both@hs-anhalt.de

E-mail: gashkov@dom.raid.ru

Abstract. Question Answering refers to the task of providing a matching answer for a given user's natural-language question. We assume here that the received response from a Question Answering system is also given as a natural-language output. From this scenario the task is raised to validate if the received natural-language answer is valid w.r.t. the given question. In this paper, we will present our approach to compute the similarity of a question to the corresponding natural-language answer while using the features of the surface form of the question and the answer. Our metrics is the distinction between a matching answer and inappropriate answers to a given question. Hence, our research agenda is dedicated to improving the answer quality of Question Answering systems by using the natural-language representation of its input and output.

1. Introduction

Answering questions in natural language plays a very important role in human-computer interactions via speech and text in all human activities. Question Answering (QA) is used to provide a reasonable answer for a given question. Common use cases are dedicated to provide facts for places, persons, etc. Typical QA systems (like Amazon Alexa, Google Assistant, Apple Siri, Yandex Alice) will provide the answer in the form of a sentence. Such systems could be implemented in smart houses as part of their improvement. Hence, for a question "What is the capital of Russia?" a QA system might answer while providing the sentence "Moscow is the capital of Russia.". However, the sentence "The capital of Russia is Moscow." is also an appropriate answer for the given question. In contrast, answers like "Moscow is a prioritization technique.", "Russia is an Australian racehorse", or "The capital of Germany is Berlin." are not appropriate answers. This paper is dedicated to the hypothesis that the surface forms of a question and its corresponding correct answers should be similar to each other. We will use this assumption in order to compute the chances of a correct or wrong answer while using the text features of questions and answers on a syntactical level. Hence, while computing this similarity, we provide a function to improve the quality of a QA system while computing an additional confidence value for the provided natural-language answer.

To validate this hypothesis we use the VANiLLa dataset containing English questions and corresponding (correct) answers. Since the VANiLLa dataset mostly consists of factoid questions, we focus on that kind of questions in our paper. To compute the similarity, the BERT classification model was used. This model is intended to classify a sentence or to decide if two sentences form a sequence. For example, the sentences "A boy goes to the shop." and "He buys apples and bread." can follow each other. If we replace their order, the sequence is broken. It is obvious that typically a question and its correct answer form such a sequence. So the BERT classifier can be used to find similar pairs of questions and answers.

The paper is structured as follows. In the next section, the related work is presented. In Section 3 the approach is presented followed by a description of the dataset. The experiment and its results are described in Section 4. The paper is concluded in Section 5.

2. Related Work

Validating the question answering quality in QA systems is not a new task. A number of studies have used different research questions for solving the QA validation problem.

Cai et al. [1] described a Chinese QA system able to find answers on the Web. This system uses the answer validation method based on combining the similarity calculation and the correlation calculation together to select one answer from a pool of candidate answers. To estimate the sentence similarity between question and answer, the authors exploit the following parameters: (1) number of the same non-repeat keywords, (2) sentence length, (3) sequence of keywords, and (4) keywords distance. The answer validation process of the system presented in this paper is decomposed into two steps: (1) choosing the best answer based on similarity, and (2) choosing the answer based on the correlation between the question and a candidate answer. If the system can find the answer by rules, the validation procedure ends. Otherwise, a second validation step is executed. In order to compare the experiment result, the authors considered the validation based on similarity as a baseline method. The precision of answers amounted to about 71%. Then the authors used the developed validation method based on both similarity and correlation. The experiment result shows that the precision of the answer reached about 76% with the total precision of about 74.3%.

Another approach of using the similarity of questions and answers is described in [2]. In this paper, a generate-validate approach was developed to answering questions about qualitative relationships, especially with respect to the QuaRel dataset [3]. The authors have shown the successful application of the generate-validate framework to solve qualitative word problems. Moreover, they have demonstrated the opportunities of transfer learning that are available in this framework. The developed model provided an improvement of 7.93% over the previous state-of-the-art QUASP+ [2].

A method of questions similarity is also applied in Community Question Answering (CQA) forums [4] or websites [5] to find out if (a variant of) the question being asked by a user has already been posed (and possibly answered) before [4]. The authors understand question similarity as the automatic task of querying and ranking semantically similar, relevant alternative questions in CQA forums. To test the model, they ran two experiments: (1) preprocessing (lowercasing, removal of stopwords and suppression of punctuation (and all combinations of them)), and (2) word meaning similarity based on different distributions (word translation probability, Word2Vec [6], fastText [7] as well as ELMo [8]). The authors also carried out an error analysis to gain insight into the differences in performance between the system set-ups. Their results showed that the combination of preprocessing method and a word-similarity metric have considerably impacted on the end results.

Sen et al. [5] assumed the QA quality modeling problems from the community support

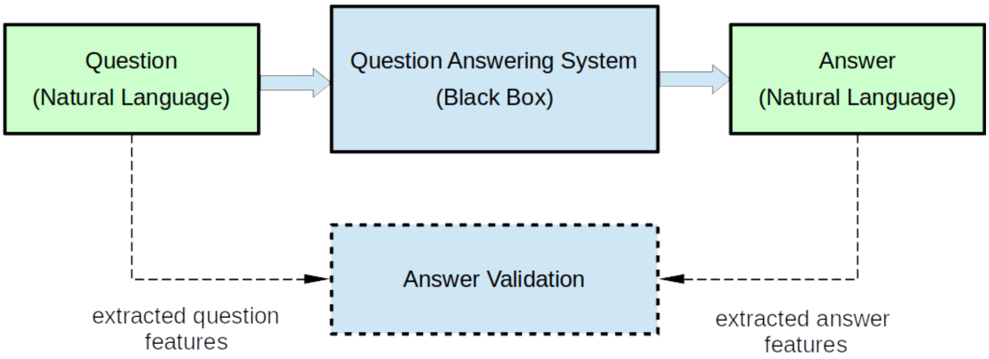


Figure 1: Data streams of black-box QA system

websites using a recently developed deep learning bidirectional transformers model. The authors proved that incorporating language semantics and technical semantics and context improves the answer quality modeling for CQA forums and websites. They investigated an entire question-answer pair for predicting the quality rather than modeling questions and answers separately. They also studied the applicability of transfer learning on QA quality modeling using Bidirectional Encoder Representations from Transformers (BERT [9]) trained on separate tasks originally using Wikipedia. The model was again pretrained and fine-tuned in the community support space (support-BERT). A general purpose language modeling framework was trained to estimate a quality of questions and answers. According to the authors, "although deep learning models may be more accurate for modeling the QA quality, due to the huge number of parameters, it is not efficient to be deployed for online question-answer quality check (or real-time question-answer quality check)" [5].

DBpedia[1] [10] is a community project to provide public access to structured linked data. The data is published following the Linked Open Data[2] initiative. An entity is identified using a URI (Uniform Resource Identifier) and described using the W3C-standardized RDF language[3]. Both entities and relations are published on the Web and can be serialized in different ways, including HTML, JSON-LD, RDF Turtle, RDF XML, and others. The English version of the DBpedia knowledge base describes over 4.5 million entities.

DBpedia as well as Wikidata[4] [11] represent knowledge graphs (KGs). In general, a QA system based on a KGs can provide answers about facts from the corresponding KG. Such systems deal particularly with factoid questions, i.e., the answer to such questions will provide facts to the user (e.g., "How many times did Bayern Munich win the Champions' League?", "What's the name of a lake in Benzie county, Michigan", "Who has Carolyne Christie as their sibling?" etc.).

To detect similarity, a number of methods can be used: n-grams [12], entity extraction [13, 14], rule-based [1, 14, 12, 2] and others. BERT [9] was used recently in many NLP tasks and demonstrated state-of-art results [5, 15, 2].

3. Approach

As described earlier, a QA system can be considered as a system providing a natural-language answer for a given natural-language question. Hence, it can be considered to be a black-box system where the input is a question and the output is one ore more answers in textual form (cf., Figure 1). We do not investigate the internal structure of a QA system but instead evaluate the

[1] https://wiki.dbpedia.org/about/
[2] https://en.wikipedia.org/wiki/Linked_data#Linked_open_data
[3] https://www.w3.org/RDF/
[4] https://www.wikidata.org/

similarity of corresponding questions and answers. By virtue of the communicative purpose – a request for information wanted – the question is naturally connected with an answer, forming together with it a question-dialog unity [16]. Therefore, the focus of the question as the nucleus of the inquiry is informational open; its function consists only in marking the focus position in the response sentence and programming the content of its filler according to the nature of ab inquiry. In a factoid question, the focus is typically expressed by a question word (e.g., "What", "Where", "When", "Which"). So the information gap between the meaning of a question and an answer is to be filled in the answer by the wanted actual information. Thus, the focus of the answer is the reverse substitute of the question word: the two makeup a focus unity in the broader question-answer construction [16]. This observation allows us to form the assumption that the question and answer are (lexically and syntactically) similar. Hence, in the simplest form of a correct answer, the question word of question sentence might just be replaced by the "right" answer (e.g., "What is the capital of Russia?" and "Moscow is the capital of Russia."). Providing a pair that is not similar (e.g., "What is the origin of association football?" and "Jan Lechabe is a politician."), we can conclude that the answer is incorrect. In some cases, the structural similarity of question and answer does not ensure that the given answer is correct. For example, the pair "What was the occupation of Bach?", "Bach was an astronaut." has a perfectly matching structure and a high similarity; however, it contradicts the reality. In such a case, common knowledge is required to detect an error, i.e., the semantics of the answer needs to be verified. Such question-answer pair cannot be addressed with our approach.

The paper is dedicated to the research question: *To what accuracy can the correct English answers be distinguished from incorrect answers while analyzing the surface forms?*

In this paper, we have used supervised machine learning to predict the similarity of a question and an answer, namely BERT [9]. Our approach is using BERT to train a binary classifier on the balanced dataset of positive and negative classes. The positive class represents a sequence of a question and a right answer, the negative one – a sequence of a question and a wrong answer. After training the classifier must predict if the question and the answer are similar or not.

4. Experiment and Evaluation

4.1. Experiment Design

The assigned task determiners the choice of the model – a binary classifier. We use BERT classifier intended to detect similarity of two sentences. BERT has a number of different sized models – small, base, large and others. Due to hardware restrictions, we selected BERT-base with 768 hidden states, 12 layers and 12 attention heads.

We use two base metrics – loss and accuracy during training. In the course of the training process, the number of epochs was set to a high value (e.g., to 1000) with the stop condition. The training is stopped if the loss on validation data shows no improvement in 5 consequent epochs. Addition metrics are calculated after the full cycle of training. The library TensorFlow [17] (version 2.3) for Python was used as a backend. TensorFlow allows to validate a model only at the epoch end, so we created a *Data Generator* to change freely the size of the epoch. After a number of experiments, the epoch's size was set to 8,000 pairs to generate enough data points for a chart.

4.2. Dataset

For the supervised training, a dataset is essential. As we chose a binary classifier, two classes are provided for training: similar and not-similar pairs. We use the VANiLLa dataset [18] that fits our needs. 98.7% of the contained questions are factoid questions which also contain a question word (e.g., "What", "Where", "When", "Which"). Some questions are asked in Jeopardy style (e.g., "Before Madden NFL, there was this electronic arts published video game.") and alternative questions (demanding a binary decision, or yes-no-questions) (e.g., “Is vidhya unni a woman or man”). Hence, the VANiLLa dataset is well-suited for the scope of the paper.

The VANiLLa dataset contains 107,166 questions with a 80% (train) – 20% (test) split. Each instance of the dataset contains 6 properties:

- `question_id`: an unique identification number for a dataset instance, e.g., 77,354
- `question`: natural-language question, e.g., “where did shanu lahiri die?”
- `answer`: retrieved answer (label of entity from Wikidata), e.g., “Kolkata”
- `answer_sentence`: a verbalization of a correct answer in natural language, e.g., “shanu lahiri died in kolkata”
- `question_entity_label`: a label of an entity, e.g., “Shanu Lahiri”
- `question_relation`: a Wikidata predicate code, e.g., “P20” (place of death)[5]

Hence, the VANiLLa dataset consists only of pairs of a question and a corresponding correct answer, i.e., out-of-the-box the data is only useful to define the training data for the positive class. In order to create also data for the negative class (non-similar question-answer pairs), we create a set of records of not related questions and answers. Our approach is as follows: while randomly selecting questions and answers from the VANiLLa dataset, we are enabled to generate a dataset where the similarity of each instance (i.e., pair of a question and an answer) has a very high probability to be pretty low. For example, consider the two (matching) pairs from the dataset:

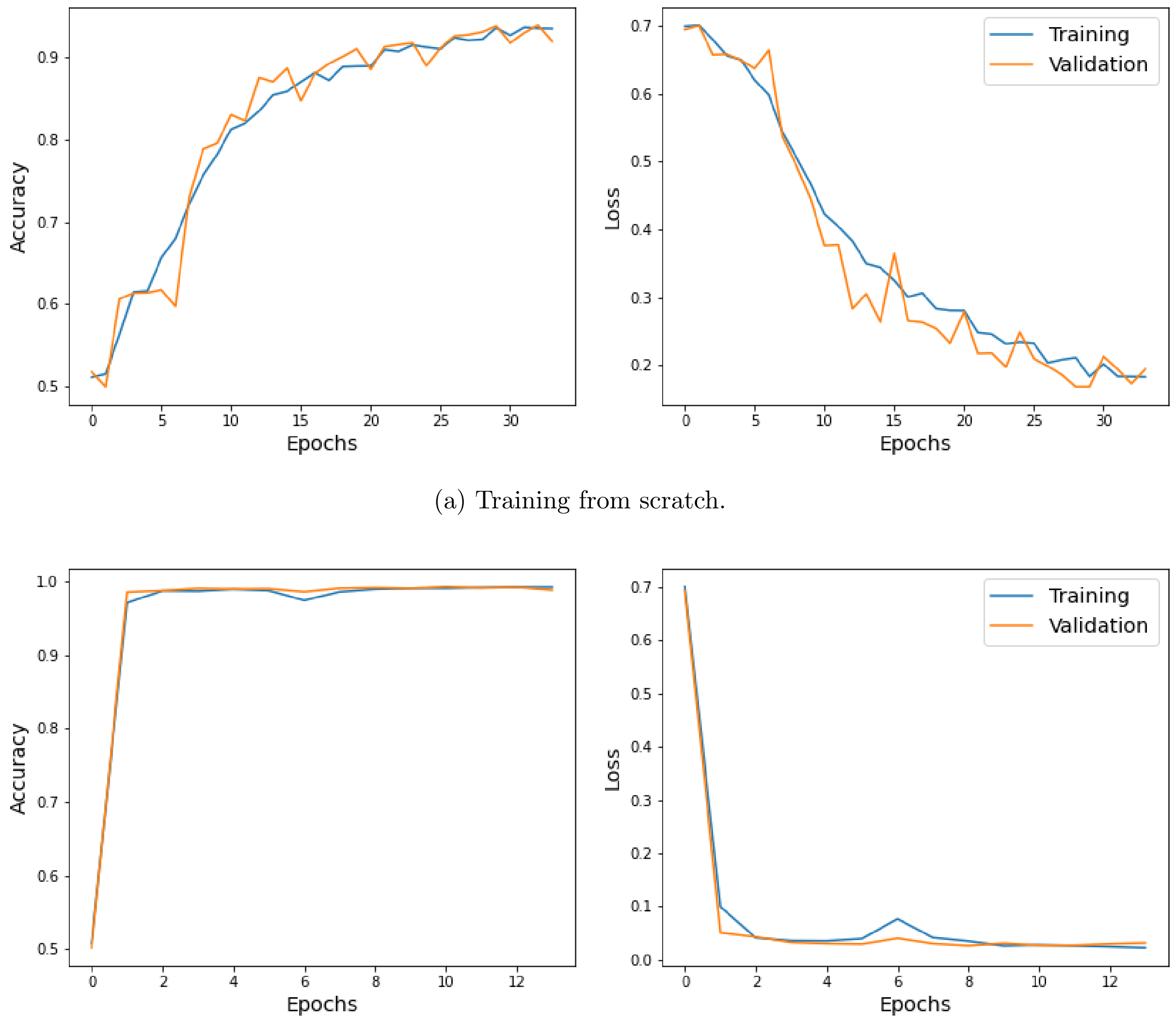
- “Where did shanu lahiri die?” and “Shanu lahiri died in kolkata.”
- “Which geographic location is Serbia situated in ?” and “Serbia is a part of europe.”

With answers exchanged, we receive two negative sequences:

- “Where did shanu lahiri die?” and “Serbia is a part of europe.”
- “Which geographic location is Serbia situated in ?” and “Shanu lahiri died in kolkata.”

This way we can easily generate negative records for the training set. A problem is that the dataset has repeating records, so uncontrolled swapping can lead to false negative pairs (e.g., the question “Which territory is the country of origin of association football?” refers to the same answer as the question “What is the country of origin of association football?”). To avoid this, we use additional information provided in the dataset – the subject entity and the predicate of the question. To address this issue, the dataset is divided into two subsets, so the questions with the identical subject and predicate occurred in the first subset and swapped all answers in this subset with the corresponding answers (i.e., with the same index) from the second part. After swapping an answer, there is still a tiny chance of getting a pair of a question and a corresponding answer. To ensure there are no such pairs more, we have checked 1% (857) of the records manually. There was one question-answer pair in the checked subset that can be assumed as similar “What is the job of Bobby Anthony Walker ?” and “He is a politician.”. So answers starting with “he”, “she” or “it” have due to their general applicability a higher chance of creating false positive records. We manually checked all instances of the dataset for such sentences and found 28 occurrences of answers starting with “he” and “she”, forming related pairs. Such records were corrected by swapping the answer with an answer from one of the neighbors manually, forming two unrelated records. Consequently, after applying these steps the negative dataset was generated with a number of records that can be assumed to represent false negative pairs with an estimated probability of less than 0.1%. We assumed this estimated error rate as acceptable. The generated complete training set contains 85,732 records both of positive

[5] I.e., pointing to the property `https://www.wikidata.org/wiki/Property:P20` of the Wikidata knowledge base.



(a) Training from scratch.

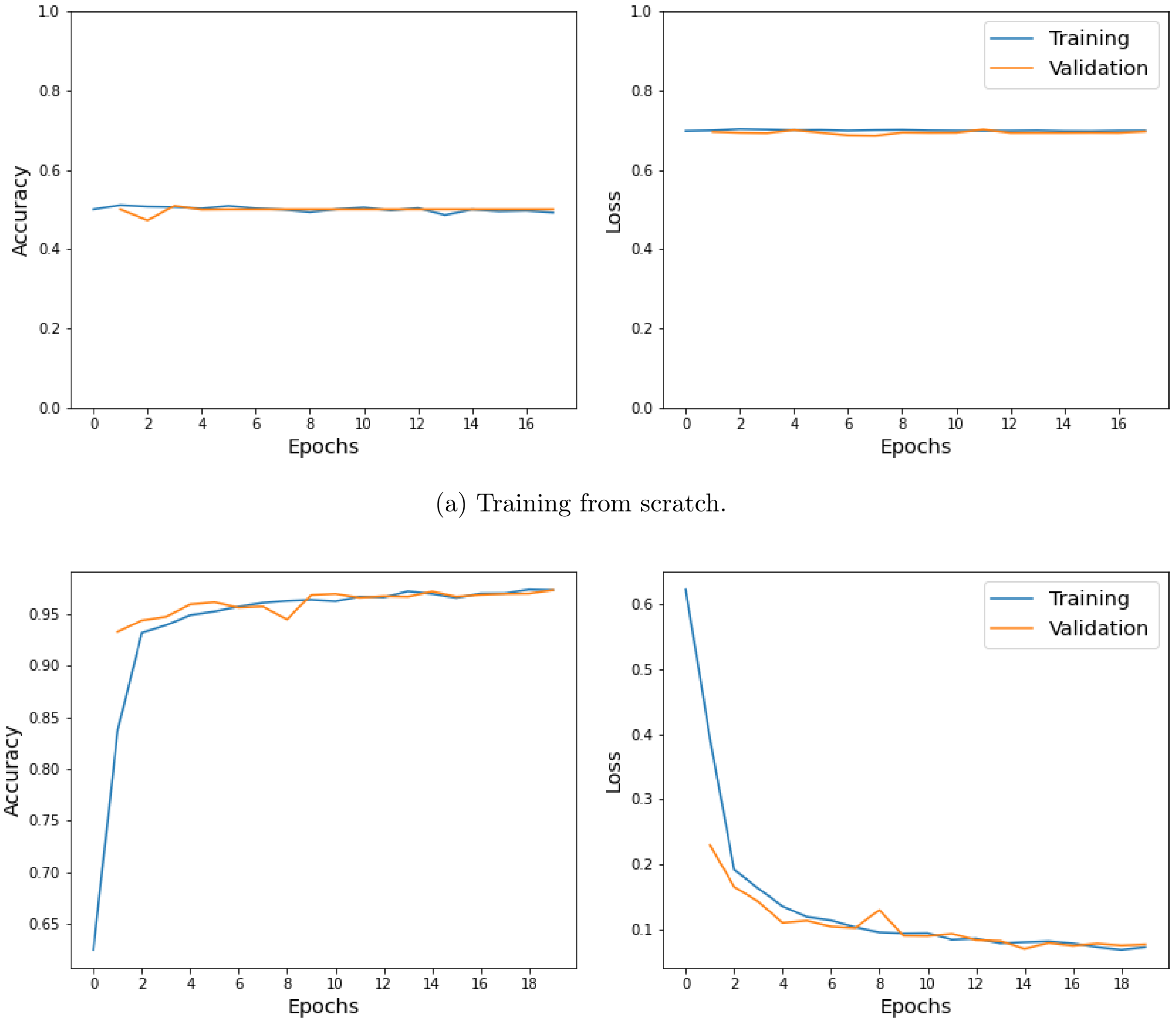
(b) Pre-trained model.

Figure 2: Training and Validation Accuracy and Loss of $Dataset_1$

and negative classes, a total of 171,464 records. The corresponding test set contains 21,434 for both classes, a total of 42,868 records. Both sets are balanced (e.g., number of positive pairs equals to the number of negative ones). We call it $Dataset_1$.

While following this approach, a test dataset was created that might not challenge the classifier due to the randomly chosen answer for a given question.

To test our approach, we create a second dataset – $Dataset_2$ which will contain more challenging question-answer pairs. This initiative is driven by the observation that in $Dataset_1$ typically both subject and predicate are different for the question and the answer. $Dataset_2$ which based on the questions from VANiLLa dataset should provide wrong answers that by a number of criteria shows a significant similarity to the question. We use the field `question_relation` of the VANiLLa dataset which refers to the Wikidata predicate used in the question. Grouping records under the same value of this field results in a set of highly similar question-answer pairs.



(a) Training from scratch.

(b) Pre-trained model.

Figure 3: Training and Validation Accuracy and Loss of $Dataset_2$

For example, for `question_relation = P495` (i.e., “country of origin”[6]) the following pairs are defined:

- “what country is four rode out from ?” and “four rode out is from spain.”
- “what is the origin of davul ?” and “turkey is the country of origin of davul.”
- “what country is heera aur pathar filmed ?” and “heera aur pathar is from pakistan.”

The process of creating negative pairs is the very same as described above, except the fact that answers were swapped only inside one predicate group.

4.3. Experimental Results

[6] cf., `https://www.wikidata.org/wiki/Property:P495`

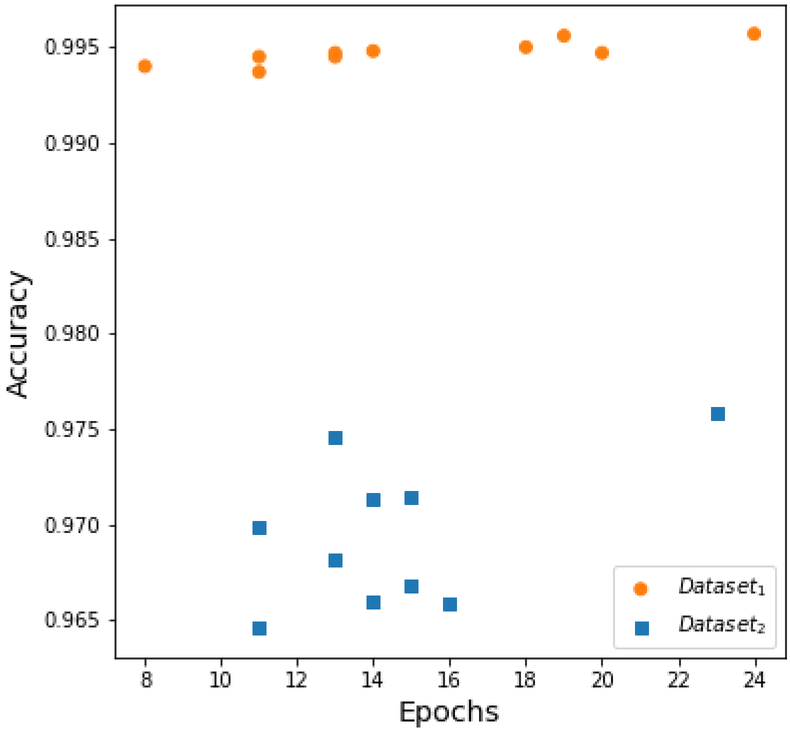


Figure 4: The maximal accuracy and epoch of predictions for $Dataset_1$ and $Dataset_2$

4.3.1. $Dataset_1$ The training and validation accuracy and loss for both fine-tuning and training from scratch BERT on VANiLLa are presented on the Figure 2. On average the trained from scratch model reaches an accuracy of 0.9706 in 29 epochs and the fine-tuned model – 0.9942 in 15.1 epochs. While accuracy is close to two the training methods, the number of errors differs about 5 times. This difference is significant for some tasks, e.g., answer validation.

$Dataset_1$ consists of "easy" training pairs, as a question and an answer differs in all parts of the sentence in most cases. For the previously mentioned example, we can extract subject, predicate, and object (in DBpedia [11] terms) as presented in Table 1.

Table 1: Question and answer example.

Type	Sentence	Subject	Predicate	Object
Question	"Which geographic location is Serbia situated in ?"	Serbia	Location	[unknown]
Answer	"Shanu lahiri died in kolkata."	Shanu Lahiri	Death place	Kolkata

4.3.2. $Dataset_2$ The size of $Dataset_2$ is 170,748 records, 85,732 questions are contained in the positive class and 85,016 in the negative one. The negative class has little less records because 1) some predicate groups have an odd number of entries and one entry was deleted to create two subsets of equal size and 2) number of predicate groups have the same subject for all entries (e.g., for predicate "P689"[7] – "afflicts" the subject is "Homo sapiens"), so it is impossible to

[7] cf., https://www.wikidata.org/wiki/Property:P689

form negative pairs. The model's test on the second question-answer set shows that it does not converge while training from scratch (cf., Figure 3). Thus, we suggest that $Dataset_2$ is harder for the classification than $Dataset_1$. Fine-tuning shows nearly same results as for "easy" dataset. Average accuracy after 10 runs is 0.9706, lower by 0.0236 than accuracy for $Dataset_1$.

4.4. Discussion

Figure 4 demonstrates the maximal accuracy of prediction and epoch when it was reached on two datasets. It is evident that the results for both datasets form two clusters. It is also significant that the average accuracy differs for $Dataset_1$ and $Dataset_2$.

Obtained results reveal that the prediction difficulty differs for two datasets while using the BERT classifier. We suppose that the designed model would demonstrate lower, but acceptable accuracy for a real QA system.

Table 2: False positive and false negative rates for $Dataset_1$ and $Dataset_2$

Metric	$Dataset_1$	$Dataset_2$
FNR	0.0075	0.0074
FPR	0.0049	0.0210

Additional metrics are shown in the Table 2. The table demonstrates that the false negative rate is stable for both datasets and the false positive rate increases significant for $Dataset_2$. This means that if the trained classifier is used for an answer validation, then the same number of right answers are marked as incorrect for both datasets, but more wrong answers are marked as correct on $Dataset_2$.

The method presented in the work has proved to be a very reliable and a state-of-art method while using linguistic feature engineering. The implemented datasets are system independent, so the proposed method can be applied to any QA system which is capable of returning an answer in the full textual form.

5. Conclusion

In this paper, we addressed the challenge of providing an excellent answer quality in QA systems. Our approach is applied on the surface forms of the given questions and answers. In other words, we used only the textual representations of the English questions and answers. Hence, it is not depending on a particular method that needs to be implemented by a QA system. Therefore, a strong benefit of our approach is its high applicability to almost any (text-driven) QA systems.

We created two datasets while transforming the VANiLLa dataset. These dataset are published[8], hey can be reused by the scientific community for further research in this field.

In this paper, BERT is used to train a binary classifier to distinguish correct answers for a given question from incorrect ones. Our experiments show a reasonable quality of the trained model. This is also true while increasing the challenge via a dataset that was created in selecting specifically similar but wrong answers. In particular, the false negative rate for $Dataset_1$ is 0.0075 and 0.0074 for $Dataset_2$. So the classifier can be used to remove wrong answers with the same efficiency on both datasets. Hence, we can conclude that using a well-trained binary classifier, we are enabled to distinguish the correct answers from the incorrect ones. This provides a significant contribution to the research community as this classifier enables researchers to use our approach as an additional feature (or even as a validator) while deciding for one question from a set of answer candidates.

[8] *https://doi.org/10.6084/m9.figshare.c.5263142.v1.*

In future work, we are going to evaluate question-answer similarity on a real Question Answering system and on different languages. We follow the intention of establishing a reusable component that can be incoportated into QA frameworks (like the Qanary framework [19]). The developed approach seems to be universal, therefore, it could be applied as a control element in smart houses and smart grids, in software for electrical engineering, as well as in telecommunication and mechatronics.

References

[1] Dongfeng Cai, Yanju Dong, Dexin Lv, Guiping Zhang and Xuelei Miao 2005 A web-based chinese question answering with answering validation *2005 International Conference on Natural Language Processing and Knowledge Engineering* pp 499–502

[2] Mitra A, Baral C, Bhattacharjee A and Shrivastava I 2019 *ArXiv* **abs/1908.03645**

[3] Tafjord O, Clark P, Gardner M, Yih W t and Sabharwal A 2019 Quarel: A dataset and models for answering questions about qualitative relationships *Proceedings of the AAAI Conference on Artificial Intelligence* vol 33 pp 7063–7071

[4] Kunneman F, Ferreira T C, Krahmer E and van den Bosch A 2019 Question similarity in community question answering: A systematic exploration of preprocessing methods and models *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)* pp 593–601

[5] Sen B, Gopal N and Xue X 2020 *arXiv* arXiv–2005

[6] Mikolov T, Chen K, Corrado G and Dean J 2013 *arXiv preprint arXiv:1301.3781*

[7] Bojanowski P, Grave E, Joulin A and Mikolov T 2016 *arXiv preprint arXiv:1607.04606*

[8] Peters M, Neumann M, Iyyer M, Gardner M, Clark C, Lee K and Zettlemoyer L 2018 Deep contextualized word representations *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* pp 2227–2237

[9] Devlin J, Chang M W, Lee K and Toutanova K N 2018

[10] Auer S, Bizer C, Kobilarov G, Lehmann J, Cyganiak R and Ives Z 2007 DBpedia: A nucleus for a web of open data *The Semantic Web* (Berlin, Heidelberg: Springer Berlin Heidelberg) pp 722–735 ISBN 978-3-540-76298-0

[11] Erxleben F, Günther M, Krötzsch M, Mendez J and Vrandečić D 2014 Introducing Wikidata to the linked data web *International semantic web conference* (Springer) pp 50–65

[12] Pakray P, Bhaskar P, Banerjee S, Pal B C, Bandyopadhyay S and Gelbukh A F 2011 A hybrid question answering system based on information retrieval and answer validation. *CLEF (Notebook Papers/Labs/Workshop)*

[13] Yang L, Ai Q, Guo J and Croft W B 2016 anmm: Ranking short answer texts with attention-based neural matching model *Proceedings of the 25th ACM international on conference on information and knowledge management* pp 287–296

[14] Ligozat A L, Grau B, Vilnat A, Robba I and Grappy A 2007 Lexical validation of answers in question answering *IEEE/WIC/ACM International Conference on Web Intelligence (WI'07)* (IEEE) pp 330–333

[15] Zhong W, Tang D, Duan N, Zhou M, Wang J and Yin J 2019 Improving question answering by commonsense-based pre-training *CCF International Conference on Natural Language Processing and Chinese Computing* (Springer) pp 16–28

[16] Blokh M 2000 *A course in theoretical English grammar* (Vysshaya shkola)

[17] Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado G, Davis A, Dean J, Devin M *et al.* 2015

[18] Dubey M Vanilla dataset https://figshare.com/articles/Vanilla_dataset/12360743 published: 2020-05-22

[19] Both A, Diefenbach D, Singh K, Shekarpour S, Cherix D and Lange C 2016 Qanary - A methodology for vocabulary-driven open question answering systems *The Semantic Web. Latest Advances and New Domains - 13th International Conference, ESWC 2016, Heraklion, Crete, Greece, May 29 - June 2, 2016, Proceedings* (*Lecture Notes in Computer Science* vol 9678) ed Sack H, Blomqvist E, d'Aquin M, Ghidini C, Ponzetto S P and Lange C (Springer) pp 625–641 URL `https://doi.org/10.1007/978-3-319-34129-3_38`