Computing, Telecommunication and Control, 2022, Vol. 15, No. 4, Pp. 64–72.
Информатика, телекоммуникации и управление. 2022. Том 15, № 4. С. 64–72.

The Seventh Conference on Software Engineering and Information Management (SEIM-2022)

Конференция по разработке программного обеспечения и управлению информацией (SEIM-2022)

Research article
DOI: https://doi.org/10.18721/JCSTCS.15405
UDC 004.852

TOKEN-WISE APPROACH TO SPAN-BASED QUESTION ANSWERING

A.A. Pismenny[1] ✉ , *E.A. Sokolov*[2]

[1] Yandex, Moscow, Russian Federation;

[2] National Research University 'Higher School of Economics', Moscow, Russian Federation

✉ pismennyy.aleks@gmail.com

Abstract. Language model pre-training has led to significant success in a wide range of natural language processing problems. It was shown that modern deep contextual language models need only a small number of new parameters for fine-tuning due to the power of the base model. Nevertheless, the statement of the problem itself makes it possible to search the new approaches. Our experiments relate to the span-based question answering, one of machine reading comprehension (MRC) tasks. Recent works use loss functions that require the model to predict start and end positions of the answer in a contextual document. We propose a new loss that additionally requires the model to correctly predict whether each token is contained in the answer. Our hypothesis is that explicit using of this information can help the model to learn more dependencies from data. Our solution also includes a new span's ranking and a no-answer examples selection scheme. We also propose approaches of accounting for information about relative positions of tokens in the dependency trees and the types of dependencies in relation to syntax-guided attention. The experiments showed that our approaches increase the quality of BERT-like models on SQuAD datasets.

Keywords: machine learning, natural language processing, question answering, machine reading comprehension, dependency parsing

Citation: Pismenny A.A., Sokolov E.A. Token-wise approach to span-based question answering. Computing, Telecommunications and Control, 2022, Vol. 15, No. 4, Pp. 64–72. DOI: 10.18721/JCSTCS.15405

Научная статья
DOI: https://doi.org/10.18721/JCSTCS.15405
УДК 004.852

ПОТОКЕННЫЙ ПОДХОД К ПОИСКУ ДИАПАЗОНА ПРАВИЛЬНОГО ОТВЕТА В ВОПРОСНО-ОТВЕТНЫХ СИСТЕМАХ

А.А. Письменный[1] ✉ , *Е.А. Соколов*[2]

[1] Яндекс, Москва, Российская Федерация;

[2] Национальный исследовательский университет «Высшая школа экономики», Москва, Российская Федерация

✉ pismennyy.aleks@gmail.com

Аннотация. Использование предобученных языковых моделей привело к значительному успеху в решении широкого круга задач обработки естественного языка. Показано, что современным глубоким языковым моделям достаточно лишь небольшого количества дополнительных параметров для дообучения, что достигается за счет мощности базовой модели. Тем не менее сама постановка задачи дообучения позволяет искать новые подходы. Наши эксперименты связаны с задачей поиска диапазона правильного ответа, одним из вариантов задачи машинного понимания прочитанного. Во многих современных работах для данной задачи используются функции потерь, которые предполагают, что модель предсказывает только позиции начала и конца правильного ответа в документе. В данной статье предложена новая функция потерь, направленная на то, чтобы модель правильно предсказывала, содержится ли каждый токен в правильном ответе. Наша гипотеза состоит в том, что явное использование этой информации может помочь модели извлечь больше зависимостей из данных. Предложенное решение также включает в себя новую схему ранжирования диапазонов и схему выбора примеров без правильного ответа. Предложены подходы к учету информации о взаимном расположении токенов в деревьях зависимостей и типах зависимостей вместе с использованием синтаксически управляемого механизма внимания. Эксперименты показывают, что предложенные подходы повышают результат для решений, основанных на модели BERT (Bidirectional Encoder Representations from Transformers), на наборах данных SQUAD (Stanford Question Answering Dataset).

Ключевые слова: машинное обучение, обработка естественного языка, вопросно-ответные системы, машинное понимание прочитанного, синтаксический разбор зависимостей

Для цитирования: Pismenny A.A., Sokolov E.A. Token-wise approach to span-based question answering // Computing, Telecommunications and Control. 2022. Т. 15, № 4. С. 64–72. DOI: 10.18721/JCSTCS.15405

Introduction

Question Answering (QA) is an important natural language understanding problem. One of examples of related tasks is span-based QA, where an answer is looked up as a sub-sequence in a contextual document. Selecting the span of the answer to a question might be used as a final step in the open-domain QA pipeline, therefore, advances in span-based QA can be applied for real-world question-answering problems [1]. Also, this task is one of the options of Machine Reading Comprehension (MRC) problem, where the task is to read the text and answer questions based on it (in this case the answering form is span selection). In this paper, we focus on a span-based QA task. Given a question and a passage containing the information for understanding, the task is to predict the contiguous span of answer text in this passage. Additional difficulty may lie in identifying those examples in which the passage does not contain a right answer (no-answer examples). We propose new a training objective and suitable answer selection scheme. Our approach is applied to a pre-trained BERT model [2]. Evaluation on challenging question answering

tasks shows than the proposed solution benefits the model performance. Our additional experiments focus on using information from dependency parsing. Our contributions are as follows:

1. We propose a new approach for span-based QA that takes into account more supervised information from the dataset.

2. We propose new approaches of accounting for information about relative positions of tokens in the dependency trees and the types of dependencies.

3. We validate our approaches on both SQuAD datasets and show that they are significantly better than BERT baseline.

Related work

Contextual language model pre-training has significantly improved performance in a wide range of NLU tasks including question-answering benchmarks. Some important examples are Embedding from Language models (ELMo) [3], Bidirectional Encoder Representations from Transformers (BERT) [2] and BERT-based approaches, such as RoBERTa [4] and ALBERT [5]. Many of these models use the self-attention mechanism proposed in Vaswani et al. [6]. For span-based QA, this mechanism can provide bidirectional cross attention between the passage and the question. Many works have been devoted to the inclusion of syntactic/semantic information for solving NLP tasks. For example, Zhang et al. [7] experimented with adding explicit syntactic constraints to the attention mechanism by focusing on ancestors in the dependency-parsing tree. Zhang et al. [8] used an additional module that processes information received from semantic role labeling.

Proposed methods

Training objective and prediction. The standard training objective function for span-based question answering is the negative sum of the log probabilities of the predicted distributions indexed by true start and end indices averaged over all the training examples (Seo et al. [9], Yu et al. [10], Devlin et al. [2], Liu et al. [4], Lan et al. [5] and others). On the one hand, the information about tokens between the start and end answer tokens is also used in these models. On the other hand, it is possible to take this information into account more explicitly. Our hypothesis is that using token-wise labels can help the model to learn more dependencies from data. Our model predicts whether each particular token is contained in the answer. The important question in this case is how to select answer on test data. The proposed solution is described below.

Similarly to Devlin et al. [2], we use a start vector $S \in \mathbb{R}^d$ and an end vector $E \in \mathbb{R}^d$ (fully connected layers over the encoder output) during fine-tuning. The start and end logits for one example are computed by

$$s = SH^T, \quad e = EH^T,$$

where $H \in \mathbb{R}^{n \times d}$ is encoder output, n is input length, d is hidden size.

We use the additional vector $B \in \mathbb{R}^d$ for binary classification task. Suppose the correct answer is spanned from i^{th} to j^{th} token. Denote logits for the new vector by $b = BH^T$. The loss function is

$$\mathcal{L} = \lambda_s \mathcal{L}_s + \lambda_e \mathcal{L}_e + \lambda_b \mathcal{L}_b,$$

where λ_s, λ_e, λ_b are hyperparameters. The components of the loss function are defined as follows

$$\mathcal{L}_s = -\log\left(softmax\left(s\right)_i\right), \quad \mathcal{L}_e = -\log\left(softmax\left(e\right)_j\right),$$

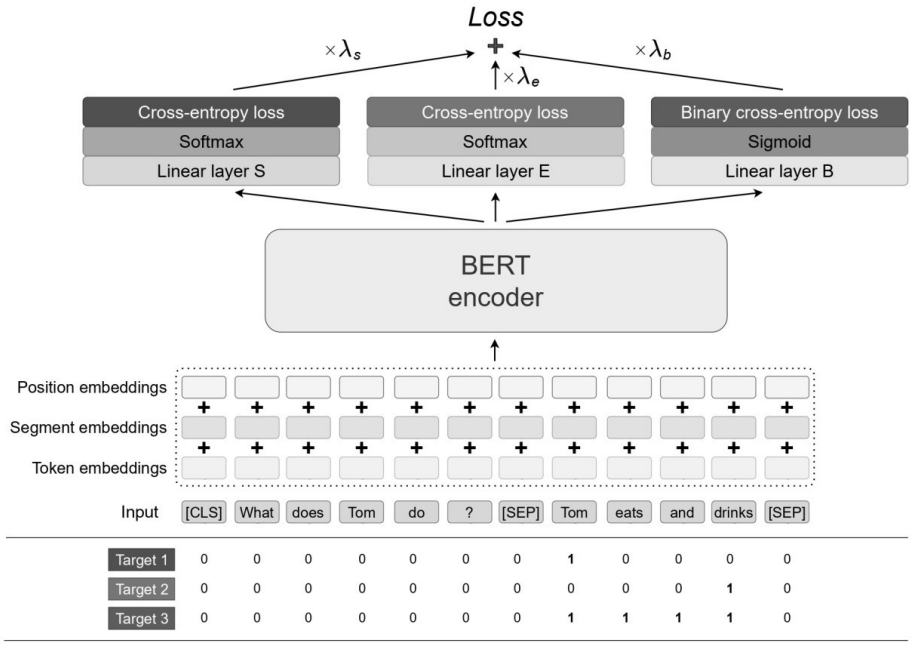

Fig. 1. DT-BERT model

$$\mathcal{L}_b = -\frac{1}{n}\left[\sum_{k\in\overline{i,j}} \log\left(\sigma\left(b_k\right)\right) + \sum_{k\in\overline{0,n-1},\, k\notin\overline{i,j}} \log\left(1-\sigma\left(b_k\right)\right)\right].$$

The DT-BERT (Double Task BERT) model is shown in Fig. 1. The score of a candidate span from position i to position j is defined as

$$T\left(i,j\right) = s_i + e_j + \frac{\alpha}{j-i+1}\sum_{k\in\overline{i,j}} b_k,$$

where α is a hyperparameter that is selected on the validation set. The maximum scoring span where $j \geq i$ is used as a prediction.

We also experimented with maximizing the following function:

$$P\left(i,j\right) = s_i + e_j + \frac{1}{n}\left[\sum_{k\in\overline{i,j}} \log\left(\sigma\left(b_k\right)\right) + \sum_{k\in\overline{0,n-1},k\notin\overline{i,j}} \log\left(1-\sigma\left(b_k\right)\right)\right],$$

which is equivalent to finding the span with the minimal value of the loss function (similarly to $s_i + e_j$ maximizing for the base model). In this case, the model performed worse. Replacing $1/n$ coefficient with a hyperparameter, which is selected on the validation set, gives comparable results.

The extra layer B can also be used to predict unanswerable questions. As usual for BERT-based models, we treat questions that do not have an answer as having an answer span with start and end at the [CLS] (null) token. We predict a non-null answer when $\max_{i\in\overline{1,n-1}} \sigma\left(b_i\right) > \sigma\left(b_0\right) + \tau$, where the threshold τ is selected on the validation set (instead of $\max_{i\leq j}\left\{s_i + e_j\right\} > s_0 + e_0 + \tau$ in [2]). For the task with unanswerable questions we first optimize the EM metric on the answerable questions by choosing α, and then we choose the optimal threshold τ on all questions for fixed α value (also for EM).

Using information from dependency parsing. We adopt the self-attention layer modification proposed in [7]. Given input token sequence $S = s_1, s_2, \ldots s_n$, where n denotes the sequence length, they derive the ancestor node set P_i for each word s_i according to the dependency tree. Then a $n \times n$ mask M is used to add explicit syntactic constraints into attention mechanism (X is layer input, for example, the batch size is 1):

$$\text{Attention}\left(Q, K, V\right) = \text{softmax}\left(\frac{QK^T + M}{\sqrt{d_k}}\right)V,$$

$$M_{i,j} = \begin{cases} 0, & \text{if } j \in P_i \text{ or } j = i, \\ -\infty, & \text{otherwise} \end{cases}$$

$$Q = W^Q X,\ K = W^K X,\ V = W^V X,\ X \in \mathbb{R}^{n \times d}.$$

We propose to supplement this mechanism with the information about the distance from each token to the root of the dependency tree corresponding to it (depth in a tree). We assume that this is sufficient information to understand the relative positions if we use described masking. The embeddings corresponding to each depth value are added to the input of dependency-based attention module:

$$Q = W^Q\left(X + D\right),\ K = W^K\left(X + D\right),\ V = W^V\left(X + D\right),$$

where $D \in \mathbb{R}^{n \times d}$. If the level of the i^{th} token is k, then $D\left[i\right] = E\left[k\right]$, where $E \in \mathbb{R}^{m \times d}$, m is the maximum depth value, E is the learnable embeddings matrix for all depths, $E\left[k\right]$ is embedding for k^{th} level.

We are experimenting with two approaches: in the first case X is BERT encoder output (similarly to [7]), in the second case X is a matrix of relations in dependency trees. Specifically, the i^{th} row of the matrix is the embedding of the dependency between the i^{th} token and its parent in the dependency tree[1].

We use special dependency embeddings for special tokens [CLS], [SEP], and [PAD]. Each of such tokens corresponds to a separate dependency tree. For proposed approach we assume that each row of Attention( ) output is some embedding of the path from the "root" of the sentence to the corresponding token. The depth embeddings and the dependency embeddings are learned together with the model. In both cases, the additional module repeats the Transformer encoder block's architecture. For the first model dependency-based module's output is averaged with the BERT encoder output with weights $w_1 = w_2 = 0.5$, for the second we use concatenation and slightly increase the dimension of the linear layers S, E and B.

Experiments

Dataset and evaluation. Our experiments are carried on SQuAD 1.1 and SQuAD 2.0 datasets [11, 12] for span-based question answering. The SQuAD 2.0 task extends the SQuAD 1.1 by allowing for the possibility that no answer exists in the provided passage. The dataset statistics are described in Table 1 and Table 2. The Dev set is another name for validation set.

Table 1

Dataset statistics of SQuAD 1.1

	Train	Dev	Test
Num of examples	87.599	10.570	9.533

[1] Most of the dependencies are covered here: https://universaldependencies.org/u/dep/all.html

Table 2

Dataset statistics of SQuAD 2.0

	Train	Dev	Test
Total examples	130.319	11.873	8.862
Examples with answer	86.821	5.928	4.530
Examples w/o answer	43.498	5.945	4.332

Two official metrics are used to evaluate the model performance on SQuAD datasets: Exact Match (EM) and a softer metric of F1 score (they were used in original papers [11, 12] and also in the official leaderboard). The EM measures the percentage of predictions that match any one of the ground truth answers exactly. The F1 score measures the overlap between the prediction and ground truth answer (for one example). The prediction and the ground truth are presented as bags of tokens, and their F1 is computed. The average F1 is calculated for all examples; in each case, the maximum F1 for all ground truth answers is taken. For unanswerable examples, EM and F1 are equal.

Implementation. We adopted the BERT-base-cased and BERT-large-cased-whole-word-masking[2] (BERT wwm) as the baselines. We used one training epoch for BERT-wwm model on SQuAD 1.1. For BERT-base-cased training, we used 3 epochs on SQuAD 1.1 and 4 epochs on SQuAD 2.0. The number of epochs was chosen from {1, 2, 3, 4} using learning rate $5 \cdot 10^{(-5)}$ to maximize the performance of our baselines. The initial learning rate was selected from $\{3 \cdot 10^{-5}, 4 \cdot 10^{-5}, 5 \cdot 10^{-5}, 6 \cdot 10^{-5}, 7.5 \cdot 10^{-5}\}$. We used the linear warmup for the first 10 % of steps followed by a linear decay to 0; L2 weight decay coefficient was 0.01. We set $\lambda_s = \lambda_e = \lambda_b = 1/3$ for proposed loss and $\lambda_s = \lambda_e = 1/2, \lambda_b = 0$ for baseline loss; our experiments with BERT-base on SQuAD 1.1 were done with double (× 2) weights λ_s, λ_e, λ_b. We used the WordPiece embeddings [13]; the maximum input length was set to 384. Our implementation was based on the Pytorch implementation of BERT[3] and Spacy dependency parser[4] (from "en_core_web_lg" model). The hidden size in dependency-based module for syntactic path encoding was set to 30, the number of attention heads was 3, and the intermediate size was 90.

Results. Table 3 shows SQuAD 1.1 Dev set results. Limited testing on this dataset is due to the fact that getting results on a test dataset is not currently available. The DT-BERT model outperforms the baselines in the task of choosing the correct span.

Table 3

SQuAD 1.1 Dev results

System	EM	F1
BERT-Base [2]	80.8	88.5
BERT-Base (our baseline)	80.94	88.67
DT on BERT-Base	**82.04**	**88.81**
BERT wwm (google-research-github[5])	86.7	92.9
BERT wwm (our baseline)	87.38	93.36
DT on BERT wwm	**87.87**	**93.54**

DT-BERT results on SQuAD 2.0 compared to prior leaderboard "BERT-base" entries and published BERT-base results are shown in Table 4.

[2] https://github.com/google-research/bert/blob/master/README.md
[3] https://github.com/huggingface/transformers
[4] https://spacy.io/models/en
[5] https://github.com/google-research/bert/blob/master/README.md

Table 4

SQuAD 2.0 results. The results of BERT on the Dev set and Test set taken from Yang et al. [14] and SQuAD 2.0 leaderboard[6] respectively

System	Dev		Test	
	EM	F1	EM	F1
BERT-Base [14]	73.66	76.30	–	–
BERT-Base (best on leaderboard)	–	–	73.10	76.24
BERT-Base (our baseline)	73.74	76.71	73.30	76.28
DT on BERT-Base	**74.48**	**77.54**	**74.77**	**77.71**

We find the proposed way to encode the relative position in dependency trees and the proposed encoding of the syntactic path useful as well.

Table 5

Results of models using dependency parsing, SQuAD 2.0

System	Dev		Test	
	EM	F1	EM	F1
DT-BERT	**74.48**	**77.54**	**74.77**	**77.71**
+ Mask layer, input:				
relation embs + depth embs	74.96	77.64	–	–
BERT output	75.20	**78.25**	–	–
BERT output + depth embs	**75.35**	78.24	**75.47**	**78.23**

We also evaluated the separate contribution of the proposed span's scoring, no-answer detection and mask layer over syntactic embeddings.

Table 6

SQuAD 2.0 Dev, ablation results. In the last version (last row), additional embeddings are not processed by the mask layer

System	EM	F1
DT-BERT	**74.48**	**77.54**
DT-BERT, w/o new no-answer detection	74.02	76.93
DT-BERT, w/o new span scoring	74.08	77.44
DT-BERT , w/o both (only new loss)	73.62	76.81
DT-BERT + Mask layer, input: relation embs + depth embs	**74.96**	**77.64**
DT-BERT + Mask layer, input: relation embs + depth embs, w/o mask layer	74.70	77.28

We used the McNemar's test [15] to test the statistical significance of SQuAD 1.1 and SQuAD 2.0 Dev set results (SQuAD test data are not publicly available). This test can be adapted for classification tasks [16], and span-based QA [17]. Our models are better than the baselines with p-value < 0.05.

[6] https://rajpurkar.github.io/SQuAD-explorer/

Conclusion

We proposed a new approach for span-based question answering task, including new loss function, span's ranking and no-answer examples selection. The experiments on two widely used benchmarks showed improving performance both for finding correct answers and for identifying unanswerable questions. The proposed solution requires a minimum number of extra parameters and can also be used for any model that encodes input sequence tokens. The solution can be further improved by using more powerful encoders or syntactic information as demonstrated. The proposed approaches to its accounting also allow improving the results.

REFERENCES

1. **Yang W., Xie Y., Lin A., Li X., Tan L., Xiong K., Li M., Lin J.** End-to-end open-domain question answering with bertserini. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics* (*Demonstrations*), 2019, Pp. 72–77.

2. **Devlin J., Chang M.-W., Lee K., Toutanova K.** Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Vol. 1 (Long and Short Papers), 2019, Pp. 4171–4186.

3. **Peters M.E., Neumann M., Iyyer M., Gardner M., Clark C., Lee K., Zettlemoyer L.** Deep contextualized word representations. *Proceedings of NAACLHLT*, 2018, Pp. 2227–2237.

4. **Liu Y., Ott M., Goyal N., Du J., Joshi M., Chen D., Levy O., Lewis M., Zettlemoyer L., Stoyanov V.** Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (*2019*).

5. **Lan Z., Chen M., Goodman S., Gimpel K., Sharma P., Soricut R.** Albert: A lite bert for self-supervised learning of language representations. *International Conference on Learning Representations*, 2019.

6. **Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser Ł., Polosukhin I.** Attention is all you need. *Advances in Neural Information Processing Systems*, 2017, Pp. 5998–6008.

7. **Zhang Z., Wu Y., Zhou J., Duan S., Zhao H., Wang R.** Sg-net: Syntax-guided machine reading comprehension. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, vol. 34, Pp. 9636–9643.

8. **Zhang Z., Wu Y., Zhao H., Li Z., Zhang S., Zhou X., Zhou X.** Semantics-aware bert for language understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, vol. 34, Pp. 9628–9635.

9. **Seo M., Kembhavi A., Farhadi A., Hajishirzi H.** Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603* (*2016*).

10. **Yu A.W., Dohan D., Luong M.-T., Zhao R., Chen K., Norouzi M., Le Q.V.** Qanet: Combining local convolution with global self-attention for reading comprehension. *International Conference on Learning Representations*, 2018.

11. **Rajpurkar P., Zhang J., Lopyrev K., Liang P.** Squad: 100,000+ questions for machine comprehension of text. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, Pp. 2383–2392.

12. **Rajpurkar P., Jia R., Liang P.** Know what you don't know: Unanswerable questions for squad. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* (vol. 2: Short Papers), 2018, Pp. 784–789.

13. **Wu Y., Schuster M., Chen Z., Le Q.V., Norouzi M., Macherey W., Krikun M., Cao Y., Gao Q., Macherey K., et al.** Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* (*2016*).

14. **Yang Z., Dai Z., Yang Y., Carbonell J., Salakhutdinov R.R., Le Q.V.** Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, 2019, Pp. 5753–5763.

15. **McNemar Q.** Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12 (1947): 153–157.

16. **Dietterich T.G.** Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10 (1998): 1895–1923.

17. **Zhang Z., Yang J., Zhao H.** Retrospective reader for machine reading comprehension. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, vol. 35, Pp. 14506–14514.

INFORMATION ABOUT AUTHORS / СВЕДЕНИЯ ОБ АВТОРАХ

Письменный Алексей Алексеевич
Alexey A. Pismenny
E-mail: pismennyy.aleks@gmail.com

Соколов Евгений Андреевич
Evgeny A. Sokolov
E-mail: sokolov.evg@gmail.com

Поступила: 30.11.2022; Одобрена: 26.12.2022; Принята: 12.01.2023.
Submitted: 30.11.2022; Approved: 26.12.2022; Accepted: 12.01.2022.