Real Life Application of a Question Answering System Using BERT Language Model

Francesca Alloatti[1,2], **Luigi Di Caro**[2] , **Gianpiero Sportelli**[1]

[1]CELI - Language Technology, Italy
[2]Department of Computer Science - Università degli Studi di Torino, Italy
{francesca.alloatti, gianpiero.sportelli}@celi.it
luigi.dicaro@unito.it

Abstract

Real life scenarios are often left untouched by the newest advances in research. They usually require the resolution of some specific task applied to a restricted domain, all the while providing small amounts of data to begin with. In this study we apply one of the newest innovations in Deep Learning to a task of text classification. The goal is to create a question answering system in Italian that provides information about a specific subject, e-invoicing and digital billing. Italy recently introduced a new legislation about e-invoicing and people have some legit doubts, therefore a large share of professionals could benefit from this tool. We gathered few pairs of question and answers; afterwards, we expanded the data, using it as a training corpus for BERT language model. Through a separate test corpus we evaluated the accuracy of the answer provided. Values show that the automatic system alone performs surprisingly well. The demo interface is hosted on Telegram, which makes the system immediately available to test.

1 Introduction

Pre-trained models have proven to be of great help in accomplishing many NLP tasks, such as natural language inference, text classification and question-answering. All of these paradigms contain a semi-supervised language model trained on large corpora of data; they are later fine-tuned to work on downstream tasks (Peters et al., 2018; Howard and Ruder, 2018; Radford et al., 2018). However, real life applications can't often benefit from these advances, for many reasons: lack of data, lack of time and resources to reach a sufficient accuracy level, or the need to address some very specific domain that elude the scope of a general-purpose architecture. As a result, many concrete scenarios of applications are left untouched by the scientific progress, even though these obstacles are far from impossible to overcome.

The goal of this study is to build a question-answering systems using only BERT (Bidirectional Encoder Representations from Transformers) language model (Devlin et al., 2018), without exploiting any rule-based refinement system or any other proprietary algorithm. This process allows to prevail over the obstacles previously listed: scarce original data was expanded mostly by using generative grammars; the whole project (data expansion plus the various training phases) took no more than eight days to complete, and the computational resources required were fairly affordable [1]. Moreover, the application domain is very specific, such that the fine-tuning of the linguistic model significantly increased the performances [2]. The architecture is simple yet effective (as shown in Figure 1) and the output of the system can be tested immediately through a Telegram bot.

2 Related Works

Since its first appearance, BERT has gained a lot of popularity in the academic community. It has been applied to various NLP tasks, including text classification for question answering. The original work by Devlin et al. (2018) contained results on BERT's performance over the Stanford Question Answering Dataset task (Rajpurkar et al., 2016), where the system had to predict the answer span for a specific question in a Wikipedia passage. Yang et al. (2019) went further, creating a question answering system deployed as a chatbot. However, both these studies tackled the task of open-domain question answering, while we focus on cases where BERT was exploited to develop systems for real life applications. For instance,

[1] CPU 8 core, GPU 28 GB, RAM 32 GB

[2] See the Results section for details on the performance.

Proceedings of the SIGDial 2019 Conference, pages 250–253
Stockholm, Sweden, 11-13 September 2019. ©2019 Association for Computational Linguistics

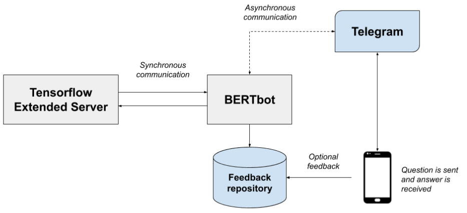
Asynchronous communication
Telegram
Synchronous communication
Tensorflow Extended Server
BERTbot
Optional feedback
Feedback repository
Question is sent and answer is received

Figure 1: Architecture of our question answering system

Lee et al. (2019) created a new BERT language model pre-trained on the biomedical field to solve domain-specific text mining tasks (BioBERT). Its results are impressive, but BioBERT is capable to perform well on domain specific knowledge because of its large pre-training process. While the pre-training surely yields better performances, it is highly expensive with regard to computational costs and time consumption. Our results show good performances even without any pre-training.

JESSI, a Joint Encoders for Stable Suggestion Inference (Park et al., 2019), was created upon the knowledge that BERT is severely unstable for out-of-domain samples. This is true for every system that does not implement any other tool other than the language model, such as ours. To solve this problem, Park et al. (2019) combined BERT with a non-BERT encoder and used a RNN classifier on top of BERT. In our case, an heuristic could be applied to the answers given by the system. It would allow to maximize the probability that the output is the correct match and not solely the one with a higher confidence score.

Other studies focus on generating pre-trained embeddings for specific domains (Beltagy et al., 2019; Alsentzer et al., 2019), but they do not test them on specific tasks.

3 BERT's Head start as a Language Model

BERT's architecture is built as a multi-layer bidirectional encoder and it is based on the Transformer model originally proposed by Vaswani et al. (2017). Although BERT has been widely used in the past year, it is not the only tool available to automatically build a working question-answering system. Attention based RNN models, especially with the addition of a LSTM or GRU module, have yielded good results on a variety of tasks (Wang et al., 2016; Zhou et al., 2016). The use of a recurrent neural network for our work was eventually ruled out for two main reasons: first, BERT encoder architecture is already trained to work as a language model on more than 104 languages (including Italian) and needs to be refined only for the specific task of text classification. The training of a RNN needs to be done for both the language model creation and the fine-tuning part, which requires a higher volume of data.

Second, the RNN training activities cannot be carried out simultaneously due to network constraints. This causes a more time-consuming and costly process.

4 Data and Fine-tuning Process

Since this aims to be a real life application, the chosen domain was e-invoicing and all the new regulations revolving around the theme of digital billing that was recently introduced by the Italian legislation. The field is very technical and specific; the data needed to reflect the features of the language employed to discuss such subject.

We first gathered pairs of clauses coherent to the domain. The data was cleaned from duplicated questions and badly written sentences, resulting in a corpus of approximately 300 pairs of sentences (a question and an answer). Half of the questions was expanded manually, while the other half - that presented recurrent linguistic patterns - was expanded using generative grammars. A grammar is written as follows:

```
{vb_might} {vb_collect} an
{n_invoice} ?
```

Resulting is sentences such as *Is it possible to collect an e-invoice?* together with all its meaningful variations. The two expansion methods created a corpus of more than 210.000 sentence pairs. No expansion was operated on the answers, since the goal is to match the correct answer to any possible expression of a question, and not to produce variegated answers.

Separately, a different corpus of 200 questions was obtained on a voluntary base from people who did not take part in the expansion process (otherwise, they would have had knowledge of the existing sentences in the training corpus). This distinct, unbiased corpus served as a test set.

4.1 The Fine-tuning

During the fine-tuning process the goal is to expand the network architecture and to train it towards a specific task. A new layer is created while the weights of the underlying original layer are modified according to the text classification job. The final training corpus consisted of 2300 sentences (obtained from the 210.000 previously mentioned). This number resulted from balancing the total of manually expanded questions with the automatically expanded ones. Otherwise we would have had an overfitting problem, since the automatic expansion generates way more sentences than the manual one. Afterwards, we used the test corpus to verify the output of the new network.

Values such as accuracy, precision and recall are not taken into consideration during the training process. Instead, the goal is to optimize, i.e. minimize, a loss function. For this study the loss function is a Cosine Proximity (1). To compute it we created a One Hot Vector that represented the 300 original sentences - each one of them as a label. The loss function takes into account two values: the One Hot Vector and the logarithm of the network output's softmax.

$$L = -\frac{y \cdot \hat{y}}{||y||_2 \cdot ||\hat{y}||_2} = -\frac{\sum_{i=1}^{N} y_i \cdot \hat{y}_i}{\sqrt{\sum_{i=1}^{N} y_i^2} \cdot \sqrt{\sum_{i=1}^{N} \hat{y}_i^2}} \quad (1)$$

Cosine Proximity Loss function

Each experimental round takes approximately one hour.

5 Results

To assess the performance, different experiments were conducted in a subsequent way to evaluate the accuracy of the test set. The first attempts were considered baseline for the following ones. When BERT model was used without applying any fine-tuning the accuracy reached 3,6 % for 40 epochs. Fine-tuning proved to be essential: accuracy on the first answer selected by the system is 86%. When considering the first three answers, the value rises up to 93,6%. The most recent experiment operates on the pre-trained language model too see if further improvement could be reached on that front. The language model was trained with new data extracted from reliable sources (operational handbooks from the *Italian Fiscal Agency*) and later fine-tuned with the same data of the previous trial. Accuracy gained +2 points, achieving 88 % on first answer.

We also compared our results to other intent matching systems such as Google DialogFlow. Using external API for intent detection accuracy reached 84%, which is slightly lower to our first experiment.

An example of the matched question (and its answer) is presented in Table 1. The user can give a feedback on each answer received, and the positive or negative feedback will add up to constantly improve the performance for the next questions. The average time to obtain a single answer is 0.2 seconds on a CPU architecture. It is therefore perfectly viable for a real time employ as a question answering system.

Unfortunately, it is impossible to compare these results with those obtained from other studies, because of the specificity of this domain, which has never been considered in this kind of experiments (at least for the Italian language).

6 Conclusion and Next Steps

We have demonstrated that it is possible to create a question answering system in a few days. The human effort was minimized - no rule was handwritten and no other algorithm was implemented - and overall the computational cost was bearable. We also showed that scarce data is not always an insurmountable obstacle, since the expansion effort can be split between manual work and automatic one. The results show that such a system can already be used with a decent degree of success. In the next future some improvements are going to be made regarding the context management and the comparison between BERT and other tools.

To improve the spectrum of questions that are correctly matched, we propose two ways to manage the dialog context using BERT:

- **External operation.** The context is given as an external factor to the model through the writing of specific rules. It modifies the labeling, i.e. the probability assigned to a label that selects a matching question.

- **Internal operation.** In this case, BERT needs to be trained towards two inputs, where one is always the context. The network changes its way of calculating the probability from *p (l|t)* (*l* being the label and *t* the text

Type of Sentence	Content
Question posed	Hello, I have a question: do I have to issue an invoice also for private clients? Even though they don't refer to any VAT number?
Question matched	Does an invoice need to be issued also towards people without a VAT number?
Answer provided	Yes, the electronic document has to be issued towards private clients without a VAT number

Table 1: Given a certain question posed by an user, the system matches one of the example in his knowledge bases and sends out the correct answer. The sentences have been translated from Italian into English for the purpose of this paper.

of the sentence) to $p\ (l|t \cap c)$.

Regarding the other tools, our goal is to verify if other models could perform equally or better given the same dataset. Many platforms are currently under review, such as Amazon Lex.

References

Emily Alsentzer, John R. Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew B. A. McDermott. 2019. Publicly available clinical bert embeddings. *arXiv preprint*, arXiv:1904.03323.

Iz Beltagy, Arman Cohan, and Kyle Lo. 2019. Scibert: Pretrained contextualized embeddings for scientific text. *arXiv preprint*, arXiv:1903.10676.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint*, arXiv:1810.04805v1.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 328–339.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *arXiv preprint*, arXiv:1901.08746v3.

Cheoneum Park, Juae Kim, Hyeon gu Lee, Reinald Kim Amplayo, Harksoo Kim, Jungyun Seo, and Changki Lee. 2019. This is competition at semeval-2019 task 9: Bert is unstable for out-of-domain samples. *arXiv preprint*, arXiv:1904.03339v1.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, , and Luke Zettlemoyer. 2018. Deep contextualized word representations. *NAACL*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *OpenAI preprint*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Lukasz Kaiser Aidan N Gomez, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Bingning Wang, Kang Liu, and Jun Zhao. 2016. Inner attention based recurrent neural networks for answer selection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1288–1297. Association for Computational Linguistics.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with bertserini. *CoRR*, abs/1902.01718.

Xinjie Zhou, Xiaojun Wanand, and Jianguo Xiao. 2016. Attention-based lstm network for cross-lingual sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 247–256. Association for Computational Linguistics.