GAAMA 2.0: An Integrated System That Answers Boolean and Extractive Questions

Scott McCarley[1], **Mihaela Bornea**[1], **Sara Rosenthal**[1],
Anthony Ferritto[2*], **Md Arafat Sultan**[1], **Avirup Sil**[1], **Radu Florian**[1]
[1] IBM Research AI
[2] AWS AI Labs
{jsmc,mabornea,sjrosenthal,arafat.sultan,avi,raduf}@us.ibm.com, ferritta@amazon.com

Abstract

Recent machine reading comprehension datasets include extractive and boolean questions but current approaches do not offer integrated support for answering both question types. We present a front-end demo to a multilingual machine reading comprehension system that handles boolean and extractive questions. It provides a YES/NO answer and highlights the supporting evidence for boolean questions. It provides an answer for extractive questions and highlights the answer in the passage. Our system, GAAMA 2.0, achieved first place on the TyDi QA leaderboard at the time of submission. We contrast two different implementations of our approach: including multiple transformer models for easy deployment, and a shared transformer model utilizing adapters to reduce GPU memory footprint for a resource-constrained environment.

Introduction

Current machine reading comprehension (MRC) systems (Alberti, Lee, and Collins 2019; Chakravarti et al. 2019; Ferritto et al. 2020) typically feature a single model targeted at supplying **short extractive** answer spans, but boolean questions demand **non-extractive YES/NO** answers, as well as supporting evidence. We demonstrate here a system that, given a question, predicts the expected answer type, provides direct YES/NO answers with supporting evidence to boolean questions, and provides short answers to extractive questions.

We highlight several capabilities, beyond those of a traditional extractive MRC system, that are necessary for our demonstration. It must be able to: 1) distinguish boolean and extractive questions, 2) generate a non-extractive YES/NO answer if the question is boolean, and 3) recognize unanswerable questions regardless of whether they produce extractive or non-extractive answers.

These new capabilities are needed because the development of MRC has been driven by training with extractive datasets (Rajpurkar et al. 2016; Kwiatkowski et al. 2019), while boolean questions have been explored in isolation (Clark et al. 2019). Therefore, extractive questions are typically handled by a pointer network which locates the start and end token of the answer span in the passage. On the other hand, boolean questions are handled by a binary classifier that classifies an entire passage with a YES or NO answer, ignoring the need to also provide concise supporting evidence for the answer. While these models are individually well-understood, we elucidate the design considerations that are necessary to present these capabilities to the user in an integrated manner. In addition, we investigate parameter sharing via adapters (Houlsby et al. 2019a) as a modeling choice to reduce the GPU memory footprint of the system in a resource-constrained environment. Our demo presents GAMMA 2.0 which adds the capability to answer both extractive *and* boolean questions in one integrated multilingual system. Other QA demos which do not support boolean questions include (Chakravarti et al. 2019; Ferritto et al. 2020; Yang et al. 2019; Yang, Fang, and Lin 2017; Zhang et al. 2021; Zhao and Lee 2020). Finally, our system backend achieves state-of-the-art results on the TyDi QA leaderboard. The performance of GAAMA 2.0 is in Table 1. Our source code is incorporated into https://github.com/primeqa/primeqa.

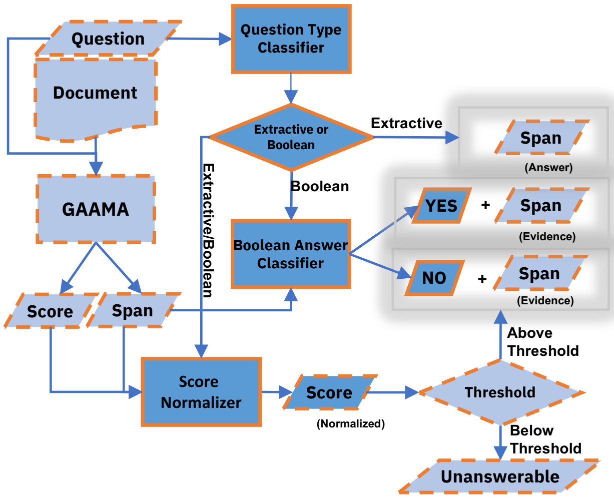

Figure 1: System diagram: Pale blue boxes are the components of a traditional MRC system; dark blue boxes are the additional components that are necessary for proper handling of both boolean and extractive questions.

[*] work completed at IBM Research AI

System Components

A diagram of the system is illustrated in Figure 1. Here we describe the components of our system.

GAAMA: A reading comprehension system that extracts a candidate answer span from a question/document pair (Ferritto et al. 2020). The model is trained with TyDi QA (Clark et al. 2020) which provides both extractive and boolean questions. Each question has a short answer and an answer passage. As in prior work, the model is trained with the short answer for extractive questions. In contrast to prior work, we also supplement with a long answer passage span as evidence for boolean questions. This enables us to produce evidence needed by the boolean answer classifier to produce a non-extractive YES/NO answer. At runtime, this component is agnostic to the difference between boolean and extractive questions, producing only a span of extracted text.

Question Type Classifier: A multi-lingual transformer-based (mBERT (Devlin et al. 2019)) classifier that takes as input the question, and returns a label that distinguishes boolean and extractive questions. It was trained and evaluated on the answerable subset of the TyDi QA questions. The classifier achieves an F1 score of $99.2/94.6$ on boolean and extractive questions respectively for TyDi QA dev.

Boolean Answer Classifier: A transformer-based binary classifier (Conneau et al. 2020) that predicts a YES or NO answer to the question. It is only invoked if the question type classifier has determined that the question is boolean. We trained the classifier using upstream system output: boolean questions from TyDi QA data, as selected by our question type classifier, along with the corresponding system output text extracted by the GAAMA component. In addition, we supplemented the TyDi QA training data with a variation of BoolQ that has extended context. Our boolean answer classifier obtains a YES/NO F1 of 91.0/44.5 on the TyDi QA dev set (NO questions are rare in TyDi QA).

Score Normalizer: A logistic regression classifier using the output of the question type classifier and the span score of the GAAMA system as features. It scales the score produced by GAAMA for the answer span to the $[0, 1]$ interval by generating a probability of whether the question/passage pair is marked as *answerable*. This score is thresholded to determine whether the question is answerable or not. The initial distribution of scores is strikingly different for boolean and extractive questions, causing many boolean questions to be unanswered. Normalization increases the percentage of answerable YES/NO questions above the threshold from 23% to 70%.

Parameter Sharing Approach

In the baseline implementation of our system, each of the transformer-based classifiers is fine-tuned independently for its particular task, and loaded in its entirety to the GPU. This implementation is convenient because the components can be developed and deployed independently, e.g. as microservices in separate docker containers. The communication between the microservices can be easily handled by the flow compiler of (Chakravarti et al. 2019). On the other hand, deploying multiple transformer-based classifiers is expensive, since GPU memory is a constrained resource.

System	Dev	System	Test
GAAMA 2.0	72.6	GAAMA 2.0	**72.35**
GAAMA	**68.6**	GAAMA-DM-Syn-ARES	68.06
		PoolingFormer	67.65

Table 1: End-to-End minimal answer F1 scores on half of the TyDi QA dev set and full test set on the official leaderboard (submitted as GAAMA-Syn-Bool-Single-Model.)

GAAMA configs	F1	# params $(\times 10^6)$	size (MiB)
Separate	72.6	1680	3204
Adapters	73.0	563	1074

Table 2: A comparison of GAMMA 2.0 using separate models and adapters. The F1 score is the minimal answer of the end-to-end system on half of the TyDi QA dev set.

To address this concern, we experiment with adapter-based models (Houlsby et al. 2019b). With this approach, there is only one transformer model, which is fine-tuned for span extraction (MRC). We implement the query type classifier and the boolean answer classifier with adapters inserted into our span extractor model, using the framework of (Pfeiffer et al. 2020). Each adapter adds $< 1\%$ additional parameters to the combined model. Only these adapter parameters are fine tuned for their respective components. The adapter-based components achieve comparable accuracy to the fully fine-tuned transformer components, and require much less GPU memory at runtime. The F1 scores are similar for both approaches, while the adapter system reduces our memory footprint significantly as shown in Table 2.

Conclusion

We present a demonstration of a machine reading comprehension system that can answer both boolean questions and factoid questions in an integrated system. When a question is boolean, it provides a direct YES/NO answer and highlights the supporting text. When a question is extractive it highlights the answer span found in the text. These new capabilities require adding additional components to a traditional MRC system: a *question type classifier*, a *boolean answer classifier* and a *score normalizer*. Each component is an essential part of a system designed to answer multiple question types. Our back-end system achieves a four point improvement over the comparable system without boolean questions and achieves state-of-the-art results on the TyDi QA leaderboard. Finally, we contrast the merits of two different implementation approaches. In one, we implement each of the components in a separate microservice for flexibility. In the other, we apply a single transformer via adapters to reduce the GPU memory footprint and associated expense.

References

Alberti, C.; Lee, K.; and Collins, M. 2019. A BERT Baseline for the Natural Questions. arXiv:1901.08634.

Chakravarti, R.; Pendus, C.; Sakrajda, A.; Ferritto, A.; Pan, L.; Glass, M.; Castelli, V.; Murdock, J. W.; Florian, R.; Roukos, S.; and Sil, A. 2019. CFO: A Framework for Building Production NLP Systems. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, 31–36. Hong Kong, China: Association for Computational Linguistics.

Clark, C.; Lee, K.; Chang, M.-W.; Kwiatkowski, T.; Collins, M.; and Toutanova, K. 2019. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2924–2936. Minneapolis, Minnesota: Association for Computational Linguistics.

Clark, J. H.; Choi, E.; Collins, M.; Garrette, D.; Kwiatkowski, T.; Nikolaev, V.; and Palomaki, J. 2020. TyDi QA: A Benchmark for Information-Seeking Question Answering in Typologically Diverse Languages. *Transactions of the Association for Computational Linguistics*, 8: 454–470.

Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; and Stoyanov, V. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 8440–8451. Online: Association for Computational Linguistics.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Ferritto, A.; Rosenthal, S.; Bornea, M.; Hasan, K.; Chakravarti, R.; Roukos, S.; Florian, R.; and Sil, A. 2020. A Multilingual Reading Comprehension System for more than 100 Languages. In *Proceedings of the 28th International Conference on Computational Linguistics: System Demonstrations*, 41–47. Barcelona, Spain (Online): International Committee on Computational Linguistics (ICCL).

Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; and Gelly, S. 2019a. Parameter-Efficient Transfer Learning for NLP. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 2790–2799. PMLR.

Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; de Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; and Gelly, S. 2019b. Parameter-Efficient Transfer Learning for NLP. In *ICML*.

Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A.; Alberti, C.; Epstein, D.; Polosukhin, I.; Kelcey, M.; Devlin, J.; Lee, K.; Toutanova, K. N.; Jones, L.; Chang, M.-W.; Dai, A.; Uszkoreit, J.; Le, Q.; and Petrov, S. 2019. Natural Questions: a Benchmark for Question Answering Research. *Transactions of the Association of Computational Linguistics*.

Pfeiffer, J.; Rücklé, A.; Poth, C.; Kamath, A.; Vulić, I.; Ruder, S.; Cho, K.; and Gurevych, I. 2020. AdapterHub: A Framework for Adapting Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020): Systems Demonstrations*, 46–54. Online: Association for Computational Linguistics.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2383–2392. Austin, Texas: Association for Computational Linguistics.

Yang, P.; Fang, H.; and Lin, J. 2017. Anserini: Enabling the Use of Lucene for Information Retrieval Research. SIGIR. ACM. ISBN 978-1-4503-5022-8.

Yang, W.; Xie, Y.; Lin, A.; Li, X.; Tan, L.; Xiong, K.; Li, M.; and Lin, J. 2019. End-to-End Open-Domain Question Answering with BERTserini. arXiv:1902.01718.

Zhang, M.; Zhang, R.; Zou, L.; Lin, Y.; and Hu, S. 2021. NAMER: A Node-Based Multitasking Framework for Multi-Hop Knowledge Base Question Answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*, 18–25. Online: Association for Computational Linguistics.

Zhao, T.; and Lee, K. 2020. Talk to Papers: Bringing Neural Question Answering to Academic Search. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 30–36. Online: Association for Computational Linguistics.