QA with Wiki: improving information retrieval *and* machine comprehension

Rohan Sampath, Puyang Ma

Document Retriever + Reader Pipeline Model (Chen et al., [2017])

Our Goal: Improve both Retriever and Reader!

Document Retriever

Original DrQA Document Retriever

- Tf-idf vectors computed for all documents and
- Documents with highest dot product with question are returned

Our Modifications

- Weighted average of tf-idf and log of PageRank score; optimum weights found to 0.5 each
- PageRank score is independent of query; it's meant to weight the retriever in favor of the most connected pages on Wikipedia

$$\mathbf{c_1 * p^T q_{doc} + c_2 * \log(PageRank_{doc} + 1)}$$

Modified Retriever does a little better on SQuAD and WebQuestions:

	SQuAD 1.1	WebQuestions	CuratedTrec
DocRetriever+bigram	69.7	66.9	30.6
Retriever+PageRank	70.1	67.5	30.6

Table 1: Results of the two retriever models on multiple datasets. % of questions for which the answer segment appears in one of the top 5 pages returned by the model.

An example of a query with better document retrieval, because the more connected ("popular") documents are returned:

```
>>> process('who is the president of the united states', 5)
+------+------------------------------------------+-----------+
| Rank |                  Doc Id                  | Doc Score |
+------+------------------------------------------+-----------+
|  1   |     Vice President of the United States  |   13.321  |
|  2   | Article Two of the United States Constitution | 13.021 |
|  3   |            President of India            |   12.664  |
|  4   |       President of the United States     |   12.548  |
|  5   |           President of Germany           |   12.196  |
+------+------------------------------------------+-----------+
```

(a) DocRetriever

```
>>> process('who is the president of the united states', 5)
+------+-------------------------------------+-----------+
| Rank |                Doc Id               | Doc Score |
+------+-------------------------------------+-----------+
|  1   |    President of the United States   |   19.556  |
|  2   |              President              |   19.065  |
|  3   | Vice President of the United States |   18.887  |
|  4   |             Sierra Leone            |   17.998  |
|  5   |            Ronald Reagan            |   17.583  |
+------+-------------------------------------+-----------+
```

(b) Modified Retriever

Figure 1: Top 5 articles retrieved to sample queries by DocRetriever and the modified retriever

Document Reader

Original DrQA Document Reader

- Paragraph tokens encoded with 300-dim GLoVe embeddings, binary feature tracking matches with question words, linguistic features (e.g., parts of speech, NER), and attention to question words
- Questions encoded based on GLoVE embeddings
- Two classifiers trained to predict start and end span based on paragraph and question input

Our Modifications

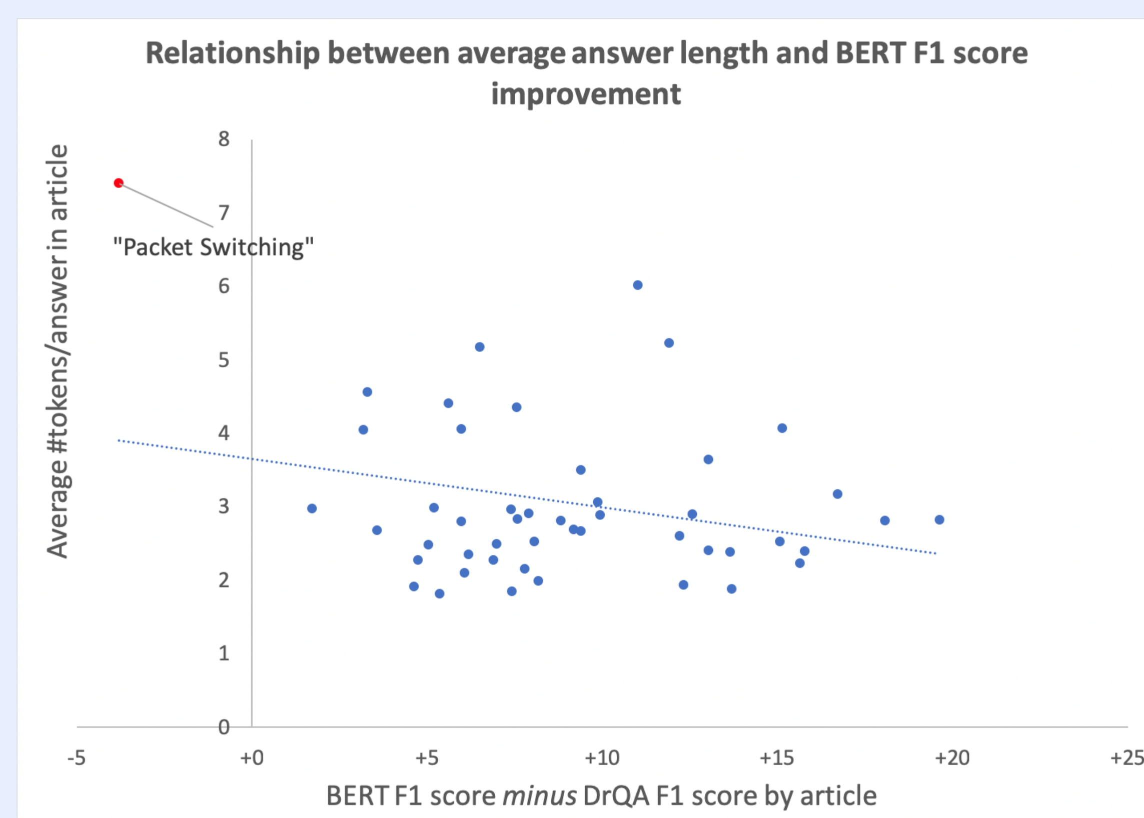
- Feature engineering (synonyms and antonyms from WordNet) doesn't improve performance by much
- Fine-tuned BERT [Devlin et al., 2018] performs very well (as expected)

Model	SQuAD 1.1 EM	SQuAD 1.1 F1
DrQA DocReader	69.3	78.6
DrQA DocReader + synonym + antonym features	69.5	78.9
BERT Reader (fine-tuned)	81.8	88.6

But… DrQA still does better on several questions (*results shown for questions in SQuAD 1.1 dev set*)

		DrQA DocReader: Correct	DrQA DocReader: Incorrect
BERT Reader	Correct	6,790 (64.2%)	1,851 (17.5%)
	Incorrect	542 (5.1%)	1,387 (13.1%)

There is some evidence to suggest that BERT produces smaller improvements on questions with long answers; more investigation required



Retriever-Reader Pipeline

Model	SQuAD 1.1 Top-2 docs	SQuAD 1.1 Top-5 docs
DrQA[2]	22.3	23.2
DrQA + PageRank Retriever + Reader with synonym, antonym features	22.5	23.4

Table 4: Best prediction exact match %, for the two pipelines on SQuAD 1.1's dev set

The Retriever-Reader "fit" score γ

$$\gamma = EM_{pipeline}/(EM_{retriever} * EM_{reader})$$

Motivation: $EM_{pipeline}$ is much lower than $EM_{retriever} \times Em_{reader}$ because: (a) Flawed $EM_{retriever}$ score, (b) DocReader trained on SQuAD, tested on Wikipedia articles, and (c) True lack of fit between Retriever and Reader (since optimization is not done across entire pipeline)

Model	SQuAD 1.1 $EM_{retriever}$	EM_{reader}	γ	$EM_{pipeline}$
DrQA	69.7	69.3	48.0	23.2
DrQA + PageRank Retriever + Reader with synonym, antonym features	70.1	69.5	48.0	23.4

*Key open question – what will **γ** be for pipeline with BERT Reader?*

Conclusions

- PageRank – limited improvement on Doc Retriever and overall pipeline performance
- BERT – significant improvement in Reader performance; some evidence to suggest that improvement gains not as high when answers are long
- Immediate next step – how does BERT affect ***γ***, and how much does it improve overall pipeline performance by?

References

[1] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. **Reading Wikipedia to answer open-domain questions**. arXiv preprint arXiv:1704.00051, 2017.

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

[3] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. **SQuAD: 100,000+ questions for machine comprehension of text**. arXiv preprint arXiv:1606.05250, 2016.