CS3244-17

Reading Comprehension On Lecture Notes

Nguyen Van Hoang, Lee Pei Xuan, Kevin Leonardo, Calvin Tantio, Luong Quoc Trung, Tan Joon Kai Daniel

Project ID: CS3244-17
School of Computing, National University of Singapore

NUS

Abstract

This project explores the application of open-domain Question Answering (QA) in learning materials with a contribution of a lecture note dataset, called LNQA, annotated with question-answer pairs. Our approach is to improve the overall pipeline of lecture note reading comprehension involving context retrieving (finding the relevant slides) and text reading (identifying the correct information). Experiments show that initializing our text reader model with a pre-trained version on SQuAD significantly improve its performance on much limited lecture note dataset, comparing with both training from scratch and inferring from the pre-trained model. Narrowing down the search space by specifying departments of questions also helps improve document retriever results, thus we examine state-of-the-art sentence classifiers in predicting departments of questions.

Motivation

Recent success in QA

A system searching answering student's queries in provided lecture notes → effectively assist lecture

Document Retriever

Retrieving contexts containing candidate answers by returning top n contexts with highest similarity to given question:

$$c^* = \arg\max_c tfidf(q) \cdot tfidf(c) \quad (1)$$

Document Reader

Signals:

- Input: question q, paragraph p
- Output: best answer span

Word representations:

- Glove Word Embedding (only feature of q)
- Exact Match: 1 if p can be exactly matched to one question word, 0 otherwise
- Linguistics features: POS, NER, TF
- Aligned question embedding: Similarity between p and q

$$p_1, ..., p_m = BiLSTM(\tilde{p}_i, ...\tilde{p}_n) \quad (2)$$

$$q_1, ..., q_l = BiLSTM(\tilde{q}_i, ...\tilde{q}_l\ ) \quad (3)$$

2 independent classifiers predicting for the answer start and end:

$$P_{start}(i) \propto exp(p_i W_{sq}) \quad (4)$$

$$P_{end}(i) \propto exp(p_i W_{eq}) \quad (5)$$

Where W is the weight matrix to be trained.

We choose the best span from token i to token i_0 such that $i <= i_0 <= i + 15$ and $P_{start}(i) \times P_{end}(i_0)$ is maximized.

Modified DrQA Model

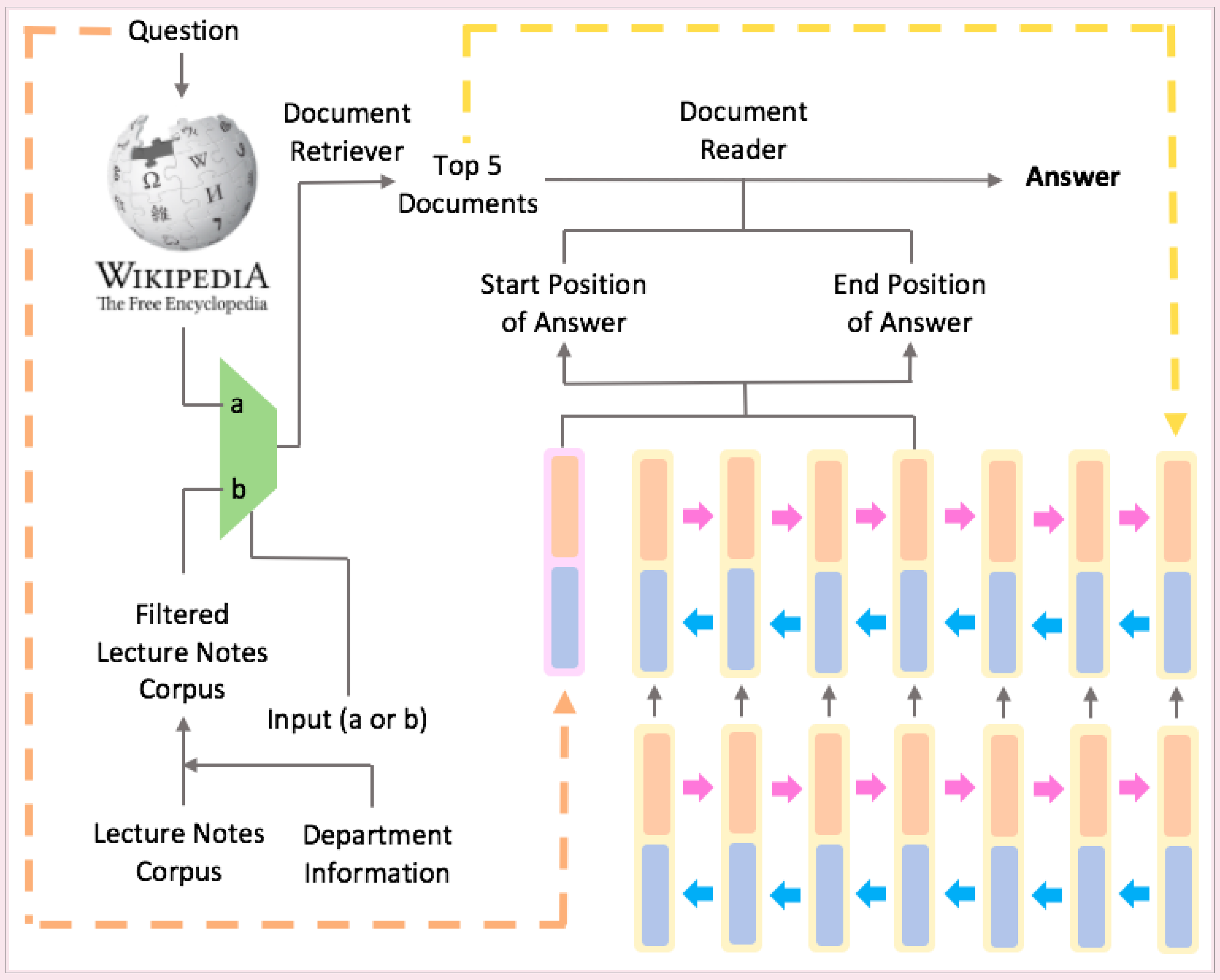

Figure 1: An overview of our modified question-answering system based on DrQA [1].

Context	Question	iQA	csQA	wsQA	sd-wsQA	Truth
estimating the prior maximum likelihood estimate mle	What does MLE represent?	likelihood estimate	estimating the prior	**maximum likelihood estimate**	**maximum likelihood estimate**	**maximum likelihood estimate**

Table 1: A sample QA pair in test set.

Hypotheses

- Due to reduced complexity and better generalization, training on LNQA from scaled-down warm-start model (sd-wsQA) on SQuAD improves reader performance compared to direct inference model (iQA), cold-start model (csQA), and full warm-start model (wsQA)
- Narrowing search space by specifying departments improves retriever performance.
- A SOTA sentence classifier (fastText) can obtain relatively good performance in department prediction on question.

Approach & Results

LNQA dataset:

- Outsourcing the data gathering process to the public using MTurk

Department specification on Document Retriever - Experiments:

- Retrieving w/ and w/o department
- fastText (SOTA sentence classifier)

Data-set	Dept	Rec@1	Rec@5
SQuAD		0.74	0.91
LN-test		0.93	0.98
LN		0.81	0.97
LN	✓	0.91	0.98

Table 2: Recall@k of retriever on different datasets.

Approach & Results (cont.)

Classifier	Data-set	Rec@1	Rec@5
fastText	LN	0.7231	0.86

Table 3: Recall@k of question classifier

Transfer learning on Document reader - Experiments:

- Dataset: SQuAD(S), LNQA(L)

Model	Pre	Train	Test	EM	F1
DrQA		S	S	69.5	78.8
sdDrQA		S	S	62.9	72.6
iQA		S	L	13.5	43.3
csQA		L	L	9.9	41.6
wsQA	DrQA	L	L	26.1	56.2
sd-wsQA	sdDrQA	L	L	**28.7**	**56.9**

Table 4: Exact match and F1 scores of different QA models.

Hyper-parameter tuning:

- Grid search (GS)
- Tree-structured Parzen Estimator (TPE)

Method	Time	EM	F1
GS	3d, 20h	24.4	55.22
TPE	<1d	28.7	57.79

Table 5: Performance of different tuning methods.

Analysis

Macroscopic level - Retriever (Table 2):

- Tf-idf retriever performs **worse** on larger datasets (Rec@1: 0.93 on LN-test, 0.81 on LN, 0.74 on SQuAD)
- Specifying department of queries **improve** retrievers (0.81 Rec@1 vs. 0.91 Rec@1) → **beneficial** to build a department classifier on questions in the long run to eliminate the need to input the department.
- Baseline SOTA classifier fastText could only achieve 0.72 Rec@1 while requiring approx. 0.9 Rec@1 to outperform baseline retriever without department → improve by adding more data or exploring other classification methods.

Macroscopic level - Reader (Table 4):

- Warm-start models (wsQA, sd-wsQA) **outperforms** cold-start and direct inferring models (csQA, iQA) → **beneficial** to initialized training on smaller dataset (LNQA) with pre-trained models on larger dataset (SQuAD)
- Scaled-down model (sd-wsQA) **outperforms** full model (wsQA) → beneficial to scaled-down the originally complex model trained on larger dataset (SQuAD) for better generalization on smaller dataset (LNQA)
- Sequential model-based optimization approach (TPE) **improves** hyper-parmeter tuning on both speed and performance - Table 5

Microscopic level - Reader (Table 1):

- Both warm-start models (wsQA and sd-wsQA) give best results in the sample test point.

Contributions

- Constructing LNQA - a QA dataset on lecture notes
- Examine transfer learning from pre-trained QA model on larger dataset (SQuAD) to a smaller dataset (LNQA)
- Examine improvement of context retrieval when specifying the departments of questions
- Examine improvement of document reader when performing Grid Search hyper-parameter tuning
- Examine improvement of time taken for hyper-parameter tuning when using Tree-Structured Parzen Estimator

References

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

Acknowledgements

We would like to extend our gratitude to the CS3244 teaching team for the opportunity to embark on this project, our anonymous reviewers for their invaluable feedback, and our Mechanical Turk respondents for their great work in our HITs. Special thanks go out to MIT OpenCourseWare for making education materials openly accessible, without which LNQA would not have been built.