Conversational Question Answering System

6759-A-H22 - Machine Learning Projects

Guided by: Alex Hernandez-Garcia

Presented by:

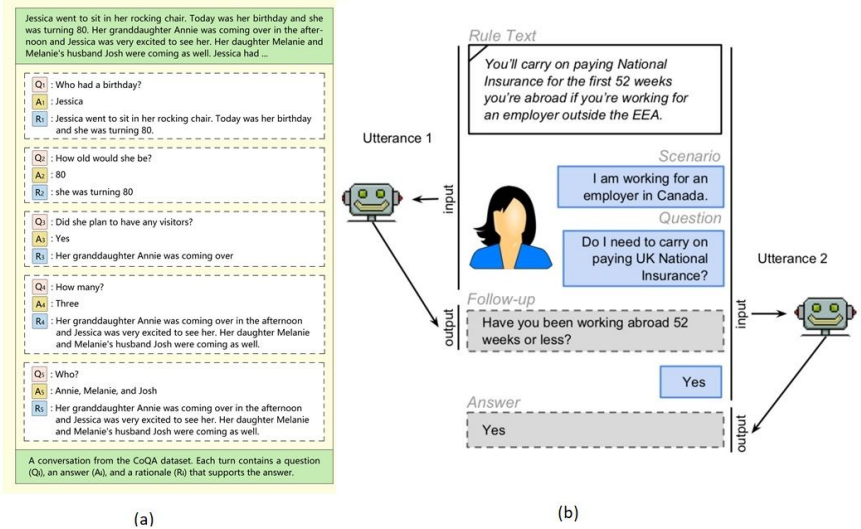
Charmi Chokshi (charmi.chokshi@umontreal.ca)
Hena Ghonia (hena.ghonia@umontreal.ca)
Sandeep Kumar (sandeep.kumar.1@umontreal.ca)
Vamsikrishna Chemudupati (vamsikrishna.chemudupati@umontreal.ca)

Agenda

- What is Conversational QA?
- What is CoQA Dataset?
- Methods explored
- Individual contribution
- Results
- Conclusions
- References

What is Conversational QA?

- Humans gather information through conversations involving a series of interconnected questions and answers.
- For machines to assist in information gathering, it is therefore essential to enable them to answer conversational questions



What is CoQA Dataset?

- CoQA is a large-scale dataset for building **Co**nversational **Q**uestion **A**nswering systems.
- The goal of the CoQA challenge is to measure the ability of machines to understand a text passage and answer a series of interconnected questions that appear in a conversation.
- The dataset contains 127k questions with answers, obtained from 8k conversations about text passages from seven diverse domains. The questions are conversational, and the answers are free-form text

What is CoQA Dataset?

The Virginia governor's race, billed as the marquee battle of an otherwise anticlimactic 2013 election cycle, is shaping up to be a foregone conclusion. Democrat Terry McAuliffe, the longtime political fixer and moneyman, hasn't trailed in a poll since May. Barring a political miracle, Republican Ken Cuccinelli will be delivering a concession speech on Tuesday evening in Richmond. In recent ...

Q_1: What are the candidates **running** for?
A_1: Governor
R_1: The Virginia governor's race

Q_2: **Where**?
A_2: Virginia
R_2: The Virginia governor's race

Q_3: Who is the democratic candidate?
A_3: **Terry McAuliffe**
R_3: Democrat Terry McAuliffe

Q_4: Who is **his** opponent?
A_4: **Ken Cuccinelli**
R_4 Republican Ken Cuccinelli

Q_5: What party does **he** belong to?
A_5: Republican
R_5: Republican Ken Cuccinelli

Q_6: Which of **them** is winning?
A_6: Terry McAuliffe
R_6: Democrat Terry McAuliffe, the longtime political fixer and moneyman, hasn't trailed in a poll since May

Figure 2: A conversation showing coreference chains in color. The entity of focus changes in Q4, Q5, Q6.

Methods explored

- **FlowQA:** We implemented LSTM and GRU-based Flow-QA architecture.
- **GraphFlow:** a GNN based architecture with various embeddings
- **Transformer:** BERT base and BERT large architectures

- **In total we performed 13 different experiments and found Transformer based architecture gives the best F1 score on the validation set.**

Individual Contribution

- Literature study and exploration of problem and dataset is being done by everyone
- GraphFlow - Hena
- FlowQA - Vamsi
- Transformer, BERT-base, BERT-large - Sandeep and Charmi

GraphFlow

Encoding Layer

- **Linguistic features**: POS(part of speech tagging), NER(named entity recognition), Exact matching
- **Pretrained word embeddings:** 300-dim GloVe embeddings and 1024-dim BERT embeddings

Reasoning Layer

- **kNN-style graph**: to select most most important edges from fully connected graph ->sparse graph
- **BiLSTM**: captures local dependency followed by a Gated graph neural network (GGNN- *RNN style structure*) which provides relational reasoning
- **Multihop message passing** in GGNN to capture long range dependency.

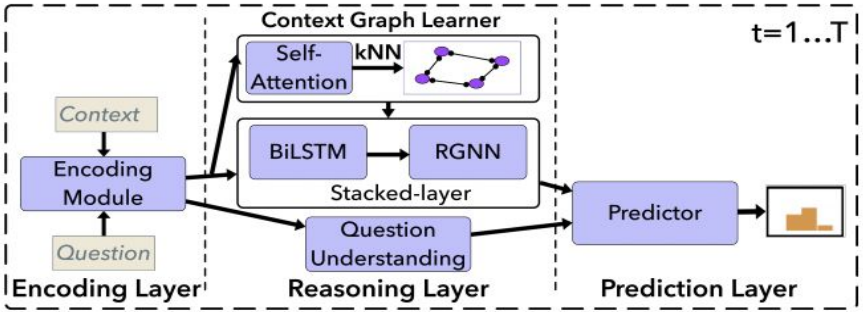


Figure 1: Overall architecture of the proposed model.

Prediction Layer

- Predicts the answer based on the matching score of question embedding and the context graph.
- Answer type classifier- to handle unanswerable questions. (e.g., “unknown”, “yes” or “no”.)

GraphFlow

- Model variants:
  - Trained model with BERT and GLoVe embedding.
    (# of parameters = 2,96,66,554)
  - Trained model only with GloVe embedding
    (# of parameters= 2,69,00,706)
  - With Bidirectional - GNN
    (performs better but computationally expensive,
    # of parameters = 3,03,87,154)

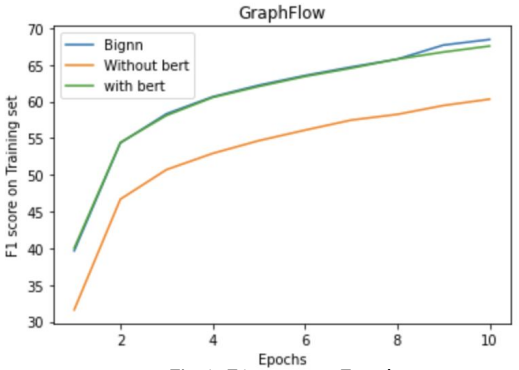


Fig 1- F1 score vs Epoch

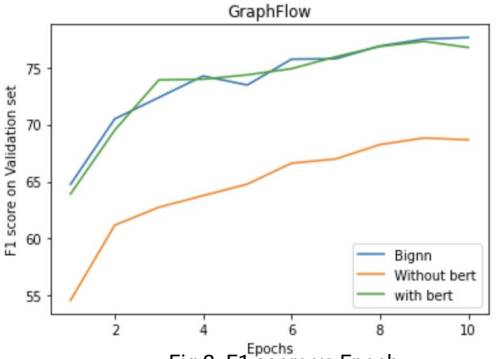


Fig 2- F1 score vs Epoch

Code : https://github.com/Hstellar/GraphFlow

FlowQA

Encoding (Question and Context):

Preprocessing the context and question data using pipeline which covers tokenization,stop words removal, POS,tagging,NER,text normalization.

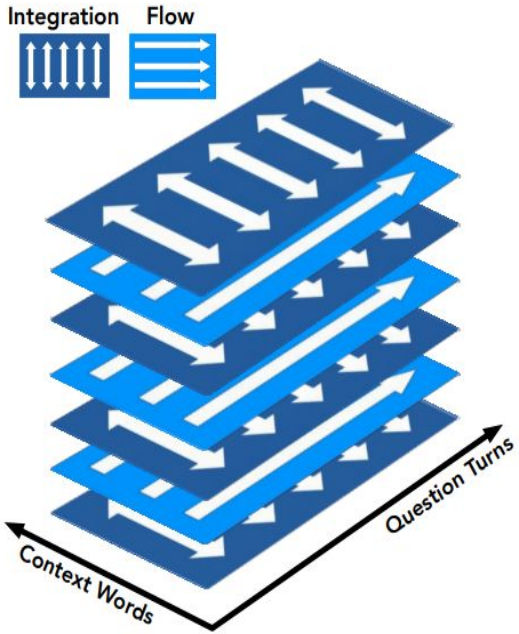
Applying Glove/Word2vec + ELMO embeddings (300 dims)

Deriving Question specific context representations for each question turn using Attention mechanism

Reasoning (Flow mechanism)

Context Integration:
The context representations calculated are passed through a BiLSTM for all questions in a parallel fashion

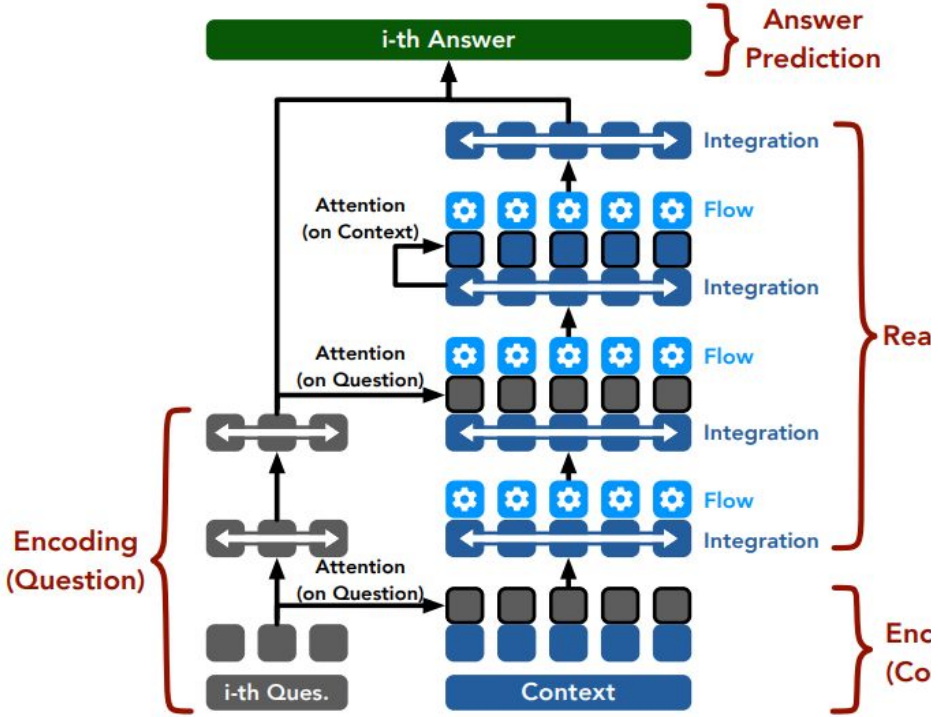
FLOW:
Intermediate representation used for answering the previous questions can be used when processing the current question



FlowQA

Fully Aware Attention mechanism:
We perform fully-aware attention on the question for each context words and on the context itself as well. It uses
History of word representations which contain word embedding, multiple intermediate and output hidden vectors in RNN

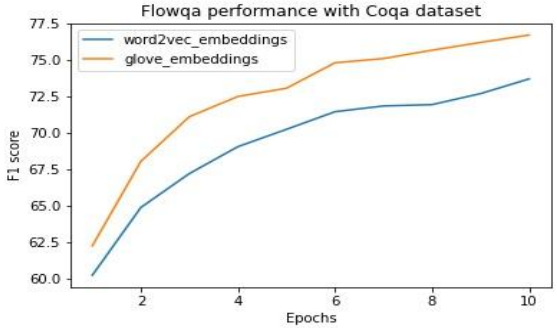
Answer prediction:
Answers are predicted by concatenating the outputs obtained from IF layers and Question specific context representations.

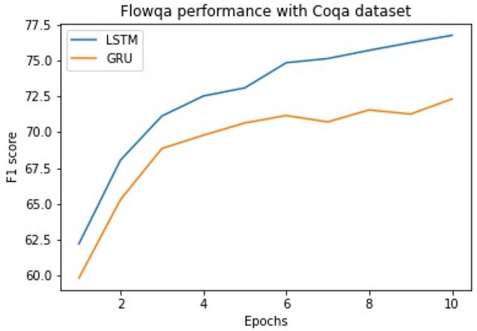


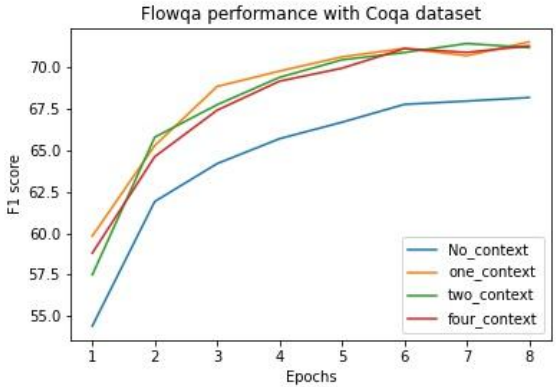
FlowQA

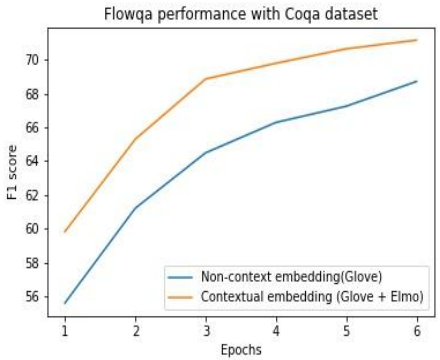
Experimentation:

1. Sequential models LSTM and GRU have been trained using the following architecture. (12M parameters)
2. Glove and word2vec methods have been used for obtaining embeddings and compared.
3. The Gru model has been tuned on the number of previous answer contexts being a part of the features while training and prediction.
4. The number of Integration flow layer have been tuned as well for the architecture.
5. The effect of contextual and non-contextual embeddings have been observed as well.









Transformer based architectures

- A transformer is a deep learning model that adopts the mechanism of self-attention, differentially weighting the significance of each part of the input data.
- It is used primarily in the fields of natural language processing and computer vision tasks.

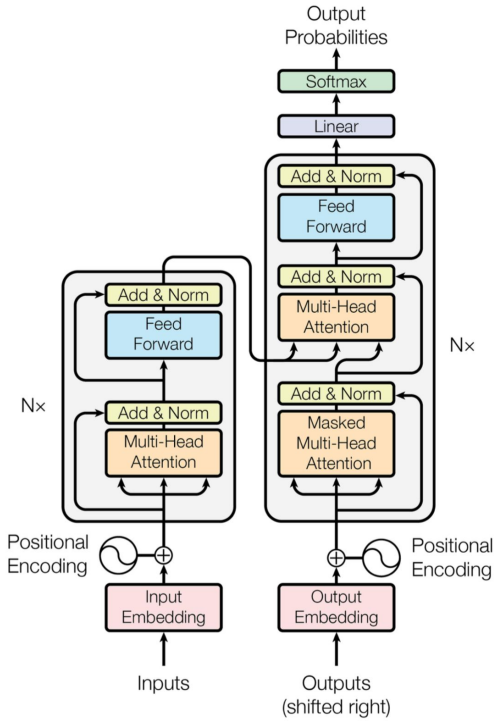
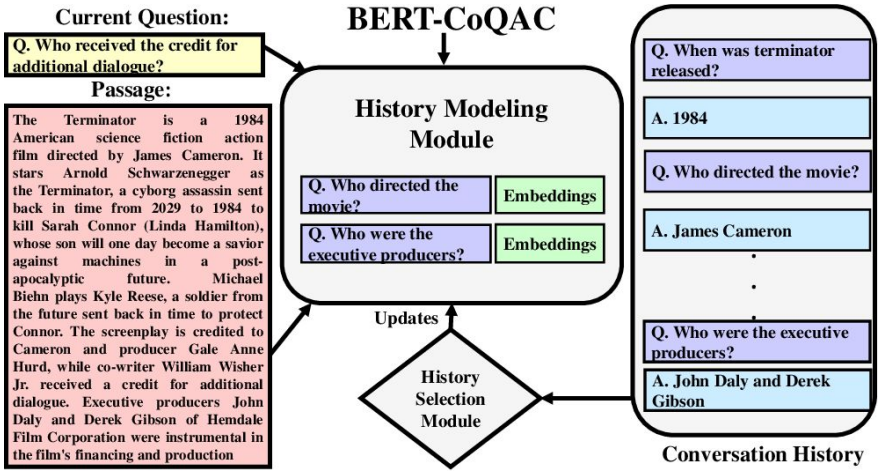


Figure 1: The Transformer - model architecture.

Transformer based architectures

- Implemented BERT-base architecture trained on lower-cased English wiki text (12-layer, 768-hidden, 12-heads, 110M parameters).
- BERT-base architecture trained on Squad data (a Question Answering data is being fine tuned which has 12-layer, 768- hidden, 12-heads, 66M parameters)
- Experimented on BERT-large model with 24-layer, 1024-hidden, 16-heads, 340M parameters.
- We tuned the history parameter to decide how many previously asked QA pairs to take into account.



Results

Model	History length	# Epochs	# Parameters	Training Time (hours)	F1 score (Validation)
FlowQA: GRU (No dialogue context)	0	8	10.5M	6	68.188
FlowQA: GRU(Glove embedding only)	2	6	8.7M	6	68.708
GraphFlow (Glove embedding)	2	10	26M	10	68.86
BERT-base (wikitext data)	0	2	110M	2	70.3
FlowQA: GRU (Glove + Elmo)	2	10	12M	6	72
FlowQA: LSTM (Word2vec+Elmo)	2	10	12M	6	73.726
Bert-large (squad data)	6	4	340M	24	76.4
GraphFlow (Glove & Bert embeddings)	2	10	29M	11	76.75
FlowQA: LSTM (Glove + Elmo)	2	10	12M	6	77
GraphFlow: BiGNN model	2	10	30M	20	77.62
Bert-base (wikitext data)	2	2	110M	2	78
Bert-base (squad data)	2	2	66M	4	78.9
Bert-large (squad data)	**2**	**2**	**340M**	**8**	**82.1**

Conclusion

- We tried various graph, flow and transformer based approaches on the CoQA dataset.
- Understood the domain of question answering & transfer learning.
- BERT based models have an immense number of parameters and hence, take a lot of time (1X Tesla V100 GPU) to be trained but have good F1 score.
- Graph based models have comparatively less no of params and training time (1 X Tesla K80/P100 GPU) but have comparatively less F1 score than BERT.
- Flow based models have the least number of parameters and hence take the least training time (1X Tesla V100 GPU). In terms of performance it is almost equivalent to the Graph based models but is not at par with BERT. The models are sequential and hence to train them faster, requires major architectural change.

Acknowledgment

We acknowledge the support of Google Developers Expert program for providing GCP credits to carry out the experiments.

References

- https://stanfordnlp.github.io/coqa/
- https://www.arxiv-vanity.com/papers/1909.03759/
- https://arxiv.org/pdf/1901.08634v3.pdf
- https://arxiv.org/abs/1810.04805
- https://arxiv.org/abs/1706.03762
- https://huggingface.co/models?pipeline_tag=question-answering
- Chen, Yu, Lingfei Wu, and Mohammed J. Zaki. "Graphflow: Exploiting conversation flow with graph neural networks for conversational machine comprehension." *arXiv preprint arXiv:1908.00059* (2019).

Questions on QA?

Thank you!