[Data Science](#)[Natural Language Processing](#)[NLP Papers Summary](#)

Day 192: NLP Papers Summary – Guiding Extractive Summarization With Question-Answering Rewards

By Ryan 10th July 2020 No Comments

Objective and Contribution

Proposed an extractive summarisation model with question-answering rewards as we believe that informative summaries should include answers to important questions. Our generated summaries yielded competitive results as measured by automatic metrics and human assessors.



Our proposed model can identify and highlight phrases that are important for answering our questions as shown below.

CNN

U.S. • Crime • Justice • Energy • Environment • Extreme Weather • Space • Science

Live TV

U.S. Edition

Former TSA agent sentenced to six months in jail for restroom recording

(CNN) **A judge this week sentenced a former TSA agent to six months in jail for secretly videotaping a female co-worker** while she was in the bathroom, prosecutors said.

During the investigation, detectives with **the Metro Nashville Police Department in Tennessee** also found that the agent, 33-year-old Daniel Boykin, entered **the woman's** home multiple times, where he took videos, photos and other data.

Police found **more than 90 videos and 1,500 photos of the victim** on Boykin's phone and computer.

The victim filed a complaint after seeing images of herself on his phone last year. [...]

Comprehension Questions (Human Abstract):
Former _____ Daniel Boykin, 33, videotaped his female co-worker in the restroom, authorities say.
Authorities say they found 90 videos and 1,500 photos of the victim on _____ and computer.

Table 1: An example extractive summary bolded in the article (top). Highlighted sections indicate salient segments useful for answering fill-in-the-blank questions generated from human abstracts (bottom).

The contributions of the paper are:

1. Proposed a novel framework of selecting consecutive words from source documents to generate extractive summaries. This involves new encoding mechanisms and sampling techniques
2. Performed empirical evaluation on information saliency by assessing summary quality with reading comprehension tasks

Methodology

Our approach is broken into four different components:

1. Extraction unit
2. Constructing extractive summary
3. Answering questions using the summary



4. Reinforcement learning

EXTRACTION UNIT

We experimented with words or chunks (phrases) as extraction units. We obtain text chunks using the sentence constituent parse tree and each chunk has at most 5 words. Note that we did not experiment with sentence level extraction like most existing work. Instead, we focused on finer-grained extraction units. We experimented with CNN and biLSTM to encode these extraction units.

CONSTRUCTING EXTRACTIVE SUMMARY

We need to identify text segments from source articles to form our extractive summary and this can be seen as a sequence labelling problem. We decided to use the framework whereby the importance of the t -th source extraction unit is determined by its informativeness, its position in the document, and the relationship with the previously selected extraction units. We have positional embeddings to encode the position of the extraction unit. At each time step, we build the vector representation of our summary up to time $t - 1$ and used it along with positional embeddings and our encoded hidden states to determine whether we should include the new extraction unit. The architecture for this is an unidirectional LSTM as shown below.

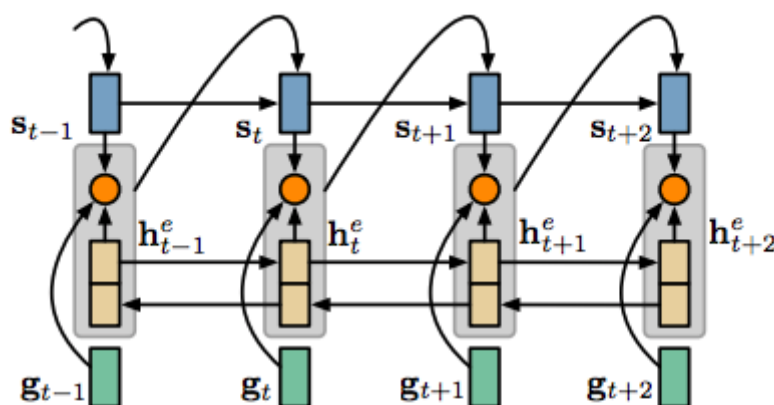


Figure 1: A unidirectional LSTM (blue, Eq. (3)) encodes the partial summary, while the multilayer perceptron network (orange, Eq. (4-5)) utilizes the text unit representation (h_t^e), its positional embedding (g_t), and the partial summary representation (s_t) to determine if the t -th text unit is to be included in the summary. Best viewed in color.

ANSWERING QUESTIONS USING SUMMARIES

To create question answer (QA) pairs, we limit our answer token to either be a salient word or named entity. We identify salient word or named entity in all the sentences in the human abstract and replace the answer token with a blank to create Cloze-style QA pair. Note that at least one QA pair should be extracted from each sentence of the abstract so that our summary includes all the useful content to answer all the questions. Overall we now have a set of QA pairs extracted from the human abstract and we can train our LSTM and attention mechanism to answer these questions using the source document.

A REINFORCEMENT LEARNING FRAMEWORK

Here, we derived a reward function that encourages our models to produce adequate, fluent, concise, and competent summaries that can perform well in our QA tasks. Our reward function has four components:

1. *QA competency*. Average log likelihood of correctly answering questions using generated summary
2. *Adequacy*. Percentage of overlap unigrams between generated and reference summary
3. *Fluency*. Encourages consecutive sequence of words to be selected
4. *Length*. Limit the summary size by setting a threshold

Experiments

Our evaluation dataset is the CNN / Daily Mail where 83% and 45% of summary unigrams and bigrams appear in source articles. We restricted the article length to 400 words and associate each article to at most 10 QA pairs to guide the extraction of summary segments. Our evaluation metric is the ROUGE scores.

COMPARISON MODELS

We compared our model with different non-neural, extractive, and abstractive models. The models include:

1. LexRank
2. SumBasic
3. KLSum
4. Lead-3



5. SummaRuNNer



6. Hierarchical with attention neural network (word and sentence based – WE and SE)

7. Distraction-M3

8. Graph attention

9. PG network + coverage

Results

We experimented with different variants of our methods. We have a baseline variant where we didn't use QA pairs during training and three other variants that uses different types of QA pairs, for example, the answer token is the SUBJ/OBJ or NER. The table 2 and 3 below showcase the results and we observed that our QASumm with different QA pairs yielded competitive results among the baseline models and outperformed the QASumm with no QA pairs variant. Our model performed at a comparable level against most SOTA results but underperformed the PG network with coverage.

We believe that extracting summary chunks rather than sentence level is key to building a concise summary but it does makes the summarisation task more challenging as the search space is larger. We also observed that the ROOT-type QA pairs have the least number of unique answers. Our QASumm + ROOT performed the best amongst the variant in daily mail dataset and QASumm + NER performed the best in CNN dataset. We suspect that training a good number of unique answers is important to maximise performance.



QA ACCURACY



In theory, an informative summary should have a high QA accuracy. We compare the summaries generated from QASumm + NoQ, the gold-standard summaries (GoldSumm), NoText (no source article), and FullText (full source article). The results are displayed below. We observed that QA with GoldSumm performed the best for all QA types, which includes FullText. This means that a highly informative summary is more useful in answering questions as searching for answers in a concise summary can be more efficient. We found that ROOT-type QA pairs can achieve high QA accuracy with NoText input which suggests that ROOT answers can be predicted using the question context. On the other hand, the NER-type QA pairs work best for FullText which most likely due to source texts containing the necessary entities to answer the questions. Therefore, we would suggest future work to include NER-based QA pairs as they encourage summaries to contain important information from the source.

EXTRACTION UNITS

We want to find out whether words or chunks are better as the extraction units. We compared the performance of our LSTM and CNN encoder and found that chunks with LSTM performed the best and chunks with CNN outperformed LSTM and CNN with words.







HUMAN EVALUATION

Each participant is given the document and three fill-in-the-blank questions. The answer tokens is chosen randomly and can be root word, the subj/obj word, or NER word. We asked the participants to rate the informativeness of the summary from 1 – 5, 5 being the most informative. We evaluated the summaries from our models and PG network. The table below showcase the average time it takes to complete a single question, the overall accuracy, and the informativeness score. Excluding human performance, our QASumm with NER-type QA pairs was able to achieved the highest accuracy and informativeness. We found that our best performing model has a wide margin in QA accuracy despite similar level of informativeness score.

Conclusion and Future Work

Our deep reinforcement learning uses a reward function (that encourages adequate  fluent summaries) to extract consecutive word sequences from source document to form  active

summary.



Source: <https://arxiv.org/pdf/1904.02321.pdf>

Ryan

Data Scientist

Previous Post

< Day 191:
Summarisation of
arXiv papers using
TextRank - Does it
work?

Next Post

Day 193: Learning
PyTorch - Tweets
Sentiment
Extraction (Part 1)





[Data Science](#) [Natural Language Processing](#) [NLP Papers Summary](#)

Day 365: NLP Papers Summary – A Survey on Knowledge Graph Embedding

Ryan

30th December 2020





[Data Science](#) [Implementation](#) [Natural Language Processing](#)

Day 364: Ryan's PhD Journey – OpenKE-PyTorch Library Analysis + code snippets for 11 KE models

Ryan

29th December 2020

