Semi-Supervised Extractive Question Summarization Using Question-Answer Pairs

Kazuya Machida*†, Tatsuya Ishigaki*†, Hayato Kobayashi‡, Hiroya Takamura†§, Manabu Okumura†
†Tokyo Institute of Technology / ‡Yahoo Japan Corporation / §AIST *equal contribution

ECIR2020

1. Introduction

Task: Extractive Question Summarization

Input : Multi-sentence question

Output : Extracted Single-sentence summary

The first sentence tends to be displayed as a headline on current CQAs, but it is not necessarily the most important one

Question: Hello, I have an AU's iPhone 5S ...	Answer: The iPhone's initial setup requires a SIM card and a PC that can use the Internet. If you don't have a PC, try connecting to Wi-Fi at a convenience store or other location. If you don't have a SIM card, borrow someone else's.
Hello, I have an AU iPhone 5S, but it still has the default settings **Default Headline Sent.** I have no Wi-Fi at home, so I cannot set it up Is there any way to do the iPhone's initial setup without Wi-Fi? **Actual Important Sent.** If there is, please tell me:)	

Our Approach: Semi-Supervised Learning

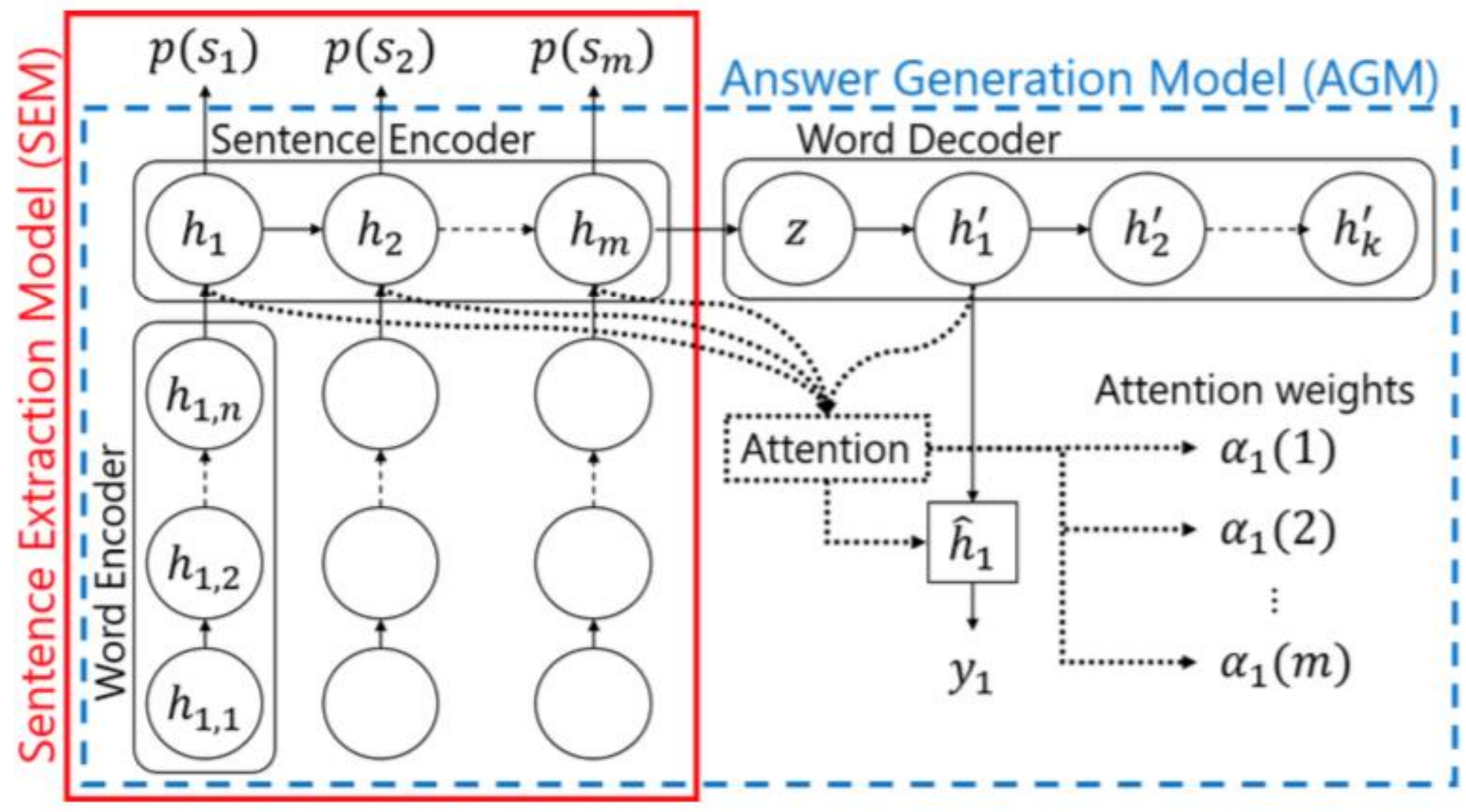
- Neural extractive summarizer requires a large labeled data, but only few labeled data exists for this task.
- We can obtain a lot of question-answer pairs.

→ We examine how to use such unlabeled paired data.

Contributions:

1. We address extractive question summarization with QA pairs as a case study of a semi-supervised setting with unlabeled paired data
2. Our experiments showed that multi-task training with an appropriate sampling method achieves better performance.
3. The data and code used in this paper are publicly available.

2. Framework



Our framework is composed of two modules:

1. Sentence Extraction Model (SEM)
   Word-level and sentence-level LSTMs convert sentences S_i into fixed-length vectors h_i. These vectors are passed on to a softmax layer to output the score $f_{ext}(s_i)$.
2. Answer Generation Model (AGM)
   LSTM-based decoder with an attention module generates an answer. We treat the averaged attention weight as score for each sentence $f_{gen}(s_i)$.

* λ, κ: hyperparameters

Training loss : $\lambda L_{\text{ext}} + (1-\lambda) L_{\text{gen}}$

Important score for s_i: $\kappa f_{\text{ext}}(s_i) + (1-\kappa) f_{\text{gen}}(s_i)$

3. Experiment

Datasets:

1. Label: Dataset with manually annotated labels (775 question)
   - We used a crowdsourcing to annotate the sentences.
2. Pair : Dataset with question-answer pairs (100K QA pairs)
3. Pseudo: Dataset with pseudo labels (2.5M sentences)
   (see another poster by us [Ishigaki+,ECIR2020]!)

Compared Models:

- Unsupervised Models
  - Lead : Simply selects the initial sentence.
  - TfIdf : Selects the sentence that has the highest Tf-Idf to the whole input.
  - SimEmb: Selects the sentence that has minimal Word Movers' Distance to the whole input.
  - LexRank: A graph-based method for sentence selection.
- Models with Label and/or Pair
  - Ext: Uses only SEM
  - Gen: Uses only AGM
  - Sep: Trains SEM and AGM separately and combine them.
  - Pre: Trains AGM first then fine-tune SEM.
  - Multi: Jointly trains AGM and SEM.
  - MultiOver: Same as Multi but Label data is oversampled.
  - MultiUnder: Same as Multi but Pair data is undersampled.
- Models with Label, Pair and/or Pseudo
  - ExtDist: Variant of Ext but trained on Pseudo data.
  - SepDist: Variant of Sep but trained on Pseudo data.
  - PreDist: Variant of Pre but trained on Pseudo data.
  - MultiDist: Variant of Multi (w/o sampling) but trained on Pseudo data.

4. Results

Accuracy = correctly selected sentences / total sentences.

* we do not use precision, recall or ROUGE since the task is a simple single-sentence extraction.

	Label	Pair	Pseudo	Acc.
Lead	-	-	-	.690
TfIdf	-	-	-	.237
SimEmb	-	-	-	.472
LexRank	-	-	-	.587
Ext	✓	-	-	.813
Gen	-	✓	-	.649
Sep	✓	✓	-	.828
Pre	✓	✓	-	.788
Multi	✓	✓	-	.770
MultiOver	✓	✓	-	.833
MultiUnder	✓	✓	-	.857
ExtDist	✓	-	✓	.838
SepDist	✓	✓	✓	.855
PreDist	✓	✓	✓	.834
MultiDist	✓	✓	✓	**.875**

- Unsupervised models do not perform well for this task.
- Multi performs well if we use an appropriate sampling.
  → Reducing data imbalance is a key factor to obtain a good performance of Multi.
- MultiDist performs the best
  → since using Pseudo data can solve the data imbalance problem by simply increasing data size.

5. Conclusion

- We proposed a framework for extractive question summarization with a semi-supervised setting.
- We found Multi-task leaning performs well if we use an appropriate sampling method.
- For future work, we will apply our framework to other tasks with similar structures, such as news articles with comments.
- The data is publicly available: http://lr-www.pi.titech.ac.jp/~ishigaki/chiebukuro/