Natural Language Understanding, Generation, and Machine Translation (2020–21)

School of Informatics, University of Edinburgh
Mirella Lapata

Tutorial 5: Summarization (Week 10)

In class we have seen how to apply the encoder-decoder model to the task of abstractive single document summarization. In this tutorial we will discuss extractive summarization and the challenges associated with it.

Question 1:

Extractive summarization, as the name implies, creates a summary by extracting sentences from the source (i.e., input) document. Most extractive models frame summarization as a classification problem. Given document D consisting of a sequence of sentences $\{s_1 \ldots, s_m\}$, we aim to select a summary from D by selecting a subset of j sentences where $j < m$. We do this by scoring each sentence within D and predicting a label $y_L \in \{0, 1\}$ indicating whether the sentence should be included in the summary.

a. We want to apply supervised training, our objective is to maximize the likelihood of all sentence labels $\mathbf{y_L} = (y_L^1, \ldots, y_L^m)$ given the input document D and model parameters θ. Can you write down this objective? sum(logP(y^i|D,theta))

b. In order to classify each document sentence as being part of the summary or not, we need to find a way to represent D in a feature space. We will use a neural network-based document encoder. Can you think of how to build this encoder in a **hierarchical** fashion? How would you represent sentences, and then the document? RNN: for each word use RNN to get embedding at sentence level then the document level.

c. So far we have managed to obtain feature representations for documents and sentences. But how do we extract sentences to create a summary? Our summarizer should label each document sentence with label 1 or 0 by estimating its **relevance** within the document. Can you think of ways to model $p(y_L|D)$? have a projection layer and softmax

d. Once you have a model that labels each sentence as positive/negative how do you create a summary? rank the sentence based on the distribution

Question 2:

How would you train the above model? In order to train our extractive model, we need ground truth in the form of sentence-level binary labels for each document, representing their membership in the summary. However, most summarization corpora only contain human written abstractive summaries as ground truth. How could you obtain binary labels from existing corpora, e.g., from CNN/Daily Mail? summary and document have the maxium overlap.

Question 3:

Can you think of why extracts might be preferable to abstracts? Can you also think of reasons why extracts might make a bad summary?

1. don't have wrong information.
2. repetitive; balance with length and important information; cohesion

Question 4:

Pretrained language models have been employed as encoders for sentence- and paragraph-level natural language understanding problems involving various classification tasks (e.g., predicting whether any two sentences are in an entailment relationship; or determining the completion of a sentence among four alternative sentences). Can you think of how you would use a pretrained language model like BERT to encode documents for extractive summarization?

cls: prediction for each sentence
relevent positioning rather than absolute positioning

interpreter the embedding