# Automatic Generation of Surveys for NLP Topics

Sally Ma
Department of Computer Science, Yale University, New Haven, CT

LILY Lab

## Introduction

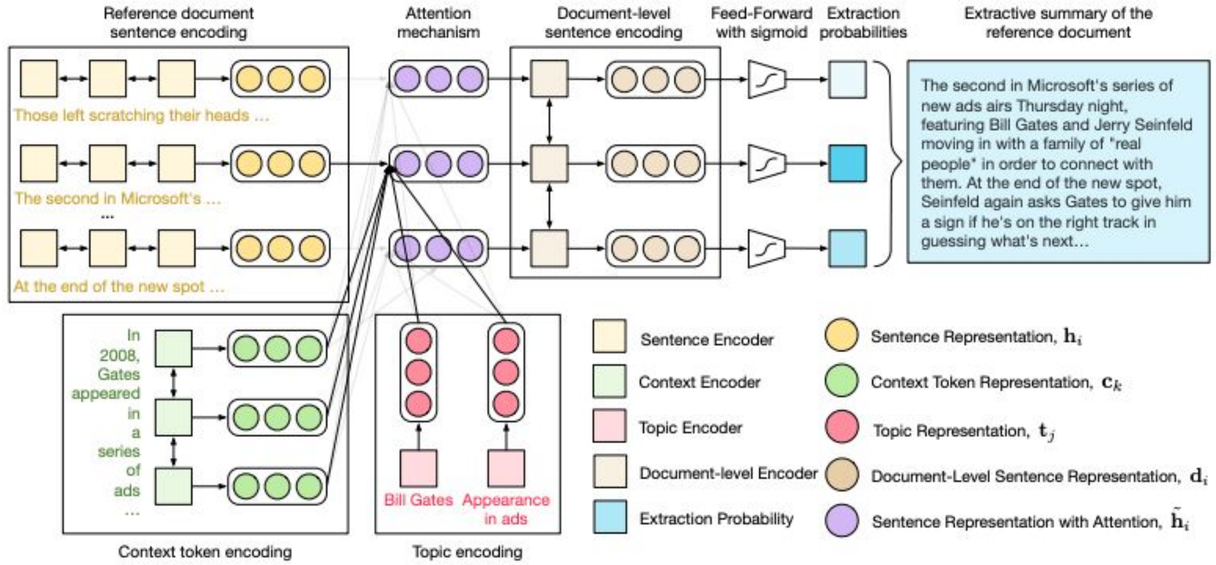
Fast-developing fields such as Artificial Intelligence (AI) often outpace the efforts of encyclopedic sources such as Wikipedia, which contain either incomplete coverage of recently-introduced topics or completely lack such content. As a result, automatically producing Wikipedia-style content is a valuable tool in order to address this information overload. We propose a two-step extractive and abstractive method and introduce a novel conditioning method applicable to other text generation tasks. We perform an extensive study on generating topic surveys for a set of Natural Language Processing (NLP) topics.

## Data

We use the WikiSum dataset, a collection of over 1.5 millions Wikipedia pages. We exclude examples that contain fewer than 3 sections because we need sufficient sections to allow for modeling interactions among the sections during training. We also exclude sections that contain information in lists or tables since our focus is on generating text-form summary.

## Approach

The first step in our two-step summarization system is extractive, where we first create section titles; we experiment with a couple approaches: 1) taking generic section titles; 2) taking section titles from related Wikipedia pages; 3) extracting from input documents (a list of references). We then create section bodies to fill the content for each section, where given a section-page title pair, we extract content relevant to the pair (grouping relevant content to each section). The second step in our system is abstractive, where we paraphrase text extracted to each section for better flow and grammar, and for avoiding copyright violation. We condition the generation upon what's already generated in previous sections to avoid repetition. For section-body creation, initial-stage research and experiments inform that we experiment with the WikiCite model proposed by Deutsch and Roth in 2019.



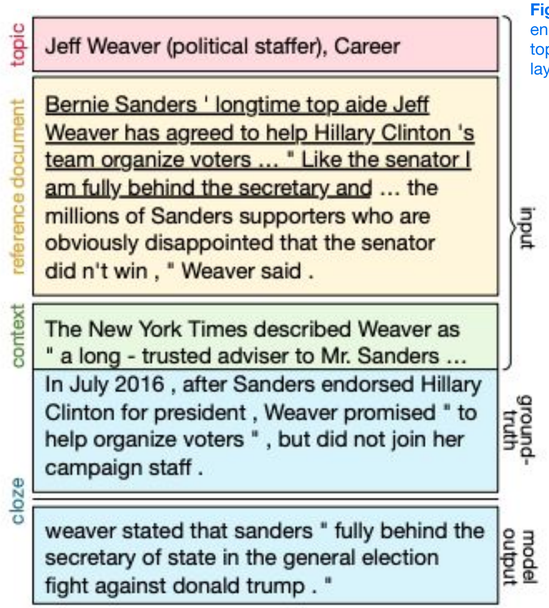




**Figure 1 (top).** Overview of the extractive model architecture: the extractive model uses 3 separate encoders to create representations for: the reference document sentences, context tokens, and topics. These are combined through an attention mechanism and passed through a feed-forward layer to compute an extraction probability for each reference document sentence.

| | Top 5 | Top 10 | Top 20 | Top 40 |
|---|---|---|---|---|
| rougeLsum-R | 0.652675 | 0.697665 | 0.735384 | 0.765089 |
| rougeLsum-P | 0.096151 | 0.080722 | 0.068139 | 0.058191 |
| rougeLsum-F | 0.147689 | 0.129112 | 0.111915 | 0.097209 |

**Figure 3 (top).** The mid score for each score type (precision, recall, and f-measure) is given for the extractive model. ROUGE-Lsum score computes longest common subsequence between each pair of reference (target) and candidate (predicted) sentences.

**Figure 4 (bottom).** Attention score array (pre-softmax) for page-section title pair ['Legislative_Council_of_Hong_Kong', 'Introduction'] after having one prediction updated for context. The first two entries are the attention weights for the topics, the rest are for the first context (which has 59 context tokens).

***[<topic 1>, <topic 2>, <context 1.1>, <context 1.2>, ... ,<context 1.59> ]***

***[1.404463 1.094372, -2.107007, -2.091571, -2.440403, ... , -2.380206, -2.296810, -1.30762]***

**Figure 2 (left).** We preprocess the Wikisum dataset to fit the format required by the model, which takes in a list of topics, a list of documents, a unique ID for each data point, an optional list of context, and a string representing the target summary The last row represents a sample output of the abstrative model.

## Model and Experimental Results

We preprocess Wikisum to fit the format required by the WikiCite extractive model (Figure 2), whose architecture is illustrated with Figure 1. For the abstractive step, we replace the list of documents with the extractive model output, and use an extension of Pointer-Generator + Coverage network as the abstractive model for outputting a paraphrased summary. Pointer-Generator network is built on seq2seq model with attention, augmented with a copy mechanism that allows the attention distribution to influence the decoder's probability distribution over vocabulary, which allows the model to copy words from input more easily. Coverage mechanism discourages attention weights from assigning high values to the same input tokens across decoding time steps repeatedly, which helps with reducing redundancy in generated summary. We experiment with both the models trained on Wikisum and the pretrained models provided by Deutsch and Roth. We also experiment with both not using and using context, which we update during inference time with what the model extracts after each prediction step. After several different modes of training and testing, the most relevant result is extraction of top 5, 10, 20, and 40 paragraphs from the Wikisum test data and computing the ROUGE-Lsum score (Figure 3). We explore attention weights for the extractive model on the test set, and find that the model properly gives high attention weights to topics (Wikipedia page and section title) and low attention weights to context (what's extracted already), as the change of section header largely determines different target summaries (Figure 4).

## Conclusion and Future Work

We present survey generation for creating Wikipedia-style surveys using a two-step extractive and abstractive method. We use the WikiSum dataset and experiment with the WikiCite model for section-body generation to fill each section, given a list of input reference documents and a pair of Wikipedia page title and section title. For future work, we plan to experiment with more models and put together the full system for final evaluation.

### Acknowledgement

I deeply appreciate the support and guidance from Alex Fabbri and Professor Dragomir Radev throughout this project.