Report: Summarization Evaluation

Kobee Raveendran

November 10, 2020

1 ROUGE Score Overview

Below are the results obtained from running the ROUGE toolkit on each of the system summaries, when compared to the human summaries from multiple experts. The full results can be found in `rouge_scores_kr.json`.

	ROUGE-1			**ROUGE-2**			**ROUGE-L**		
System	P (%)	R (%)	F (%)	P (%)	R (%)	F (%)	P (%)	R (%)	F (%)
Centroid	32.32	42.32	36.55	7.41	9.65	8.36	28.12	36.80	31.79
DPP	39.30	41.20	40.20	9.66	10.11	9.88	30.59	32.05	31.29
ICSISumm	39.24	37.17	38.16	10.12	9.54	9.82	34.50	32.67	33.54
LexRank	32.41	41.02	36.16	6.98	8.84	7.79	28.55	36.14	31.85
Submodular	38.75	39.98	39.35	9.30	9.57	9.43	33.58	34.64	34.09

Table 1: Average precision (P), recall (R), and F1 (F) scores achieved by each system on the given summaries. See the provided JSON file for all results or confirmation of the ones listed here

1.1 Experimental Setup

1.1.1 Technologies Used

Though there is not much code involved, here I describe my experimental setup. All code is written in Python (3.7) and makes use of the PyROUGE module. PyROUGE is only a Python wrapper around ROUGE, so I additionally had to install ROUGE-1.5.5 to run this. I used Ubuntu as I had some problems getting PyROUGE and ROUGE to work together on Windows. A more detailed description of my experimental setup can be found in the README.

1.1.2 Running the Experiments

Regarding ROUGE settings, I kept defaults for all options. They closely mirror the options from the homework writeup, such as `-c 95`, `-n 4`, `-m`, etc. The full ROUGE command was: `<path_to_ROUGE_perl> -e <path_to_ROUGE>/data -c 95 -2 -1 -U -r 1000 -n 4 -w 1.2 -a -m <generated_temp_folder>/rouge_conf.xml`

Runtimes were fairly quick for each system. Detailed run-time information can be found in the console output (right above each system's ROUGE scores), but on average each system was evaluated within 5.5-8 seconds.