Documentation

This project is an extended work of Mooc which attempts to improve lecture segmentation and summarization of each segment based on subtitles of the lectures.

Overview

The task of this project is reduced to sub-tasks: paragraph segmentation and keywords extraction. The approach for paragraph segmentation task uses features of each sentence in the text, and compares their similarity. The keywords extraction task summarizes a paragraph into one vector embedding and finds closest phrases to the embedded paragraph.

Implementation

The data preprocessing is done in `corpus.py` and `text_parser.py`. The `keyword_extraction` handles keyword extraction task and needed models. The `topic_modeling.py` has models for building bag-of-word topic models and sentence embedding models. And these models are used in `papragraph_segmentation` to do the papragraph segmentation task.
See tables in the later sections for all used models. All details are documented in the comments of each file.

Environment Requirements

The project is developped and tested only on `python 3.8.12`.
Here is all packages used in this project:

- `torch=1.10.0`
- `sentence-transformers=2.1.0`
- `spacy=2.3.5`
  - Installed model `en_core_web_sm`
- `pysrt=1.1.2`
- `webvtt-py=0.4.6`

(I didn't use a `requirement.txt` mainly because there is an additional model to install...)

Usage

Paragraph Segmentation

Use `paragraph_segmentation.baseline_segmentation()` to do text segmentation. The specific usege is documented in the comment.

Keywords Extraction

Use `keyword_extraction.extract_keywords()` or `keyword_extraction.extract_keywords_all()` to extract one or multiple paragraphs. The specific usege is documented in the comment.

Evaluation

Use `evaluate.evaluate_segmentation()` and `evaluate.evaluate_keyword_extraction()` to evaluate both models. The specific usege is documented in the comment.

Test Result

The entire algorithm is unsupervised, so there is no need to present training and testing error separately. And keep in mind, the testing data size is too small to yield any truly trustable result. Below is the performance of each model with untweaked parameters.

Keyword Extraction

Model	Precision	Recall
SciBERT-Nli	0.666766	0.506096
SciBERT	0.770466	0.542953
all-MiniLM-L6-v2	0.794343	0.562764

Precision is calculated by the maximum similarity between the predicted keyword and all true keywords.
And recall is calculated by the maximum similarity between the true keyword and all predicted keywords.

Paragraph Segmentation

Model	Segmentation Score

Model	Segmentation Score
LDA	3.6675
NMF	3.9311
LSI	4.0234
all-MiniLM-L6-v2	3.8637
SciBERT	4.1515

The paragraph segmentation is considered as a partition problem here. The score is the average of completeness score and adjusted MI score between predicted labels and true labels.

Contribution

Mingwei Huang (mingwei6@illinois.edu)

Here is a presentation video link.