Cutting It Short

CMPE 493 - Assignment 3

Contact: riza.ozcelik@boun.edu.tr

DEADLINE: May 4, 2019, 23:59

1 Problem Definition

In this project you will implement an unsupervised extractive single document summarization program that summarizes a BBC News article by selecting the prominent sentences from the article. To find the most prominent sentences in the article, you will use the K-Means algorithm and cluster the sentences in each article. With this scheme, the most representative sentence of each cluster can be treated as a representative sentence of the document as well. To represent the sentences in vector space, you will use pre-trained word vectors. Lastly, you will implement an evaluation scheme with k-fold cross-validation to find superior models. The dataset and the pre-trained GloVe vectors are available on this link. Note that the word vectors are provided as a dictionary pickle.

2 Implementation

The steps you will implement are as follows:

1. **Sentence Representation:** You will use pre-trained word vectors to represent words. To represent sentences, you are free to use any algorithm of your desire. Moreover, it is totally fine to use simple averaging, though more complex and robust schemes are encouraged.

2. **K-Means Clustering:** Having represented each sentence as a vector, you will cluster the sentences in a document into K clusters. How to choose K is totally up to you[1] as long as your method is applicable to the test set, i.e. unseen documents.

3. **Model Selection:** In this part, you are expected to propose at least two models and compare them with respect to a metric called *Rouge* using k-fold cross-validation and report the average Rouge-1, Rouge-2, and Rouge-L F1 scores obtained for the validation and test sets for all your proposed models. What is meant by proposing a new model is changing an open-ended part of the algorithm such as sentence representation from word vectors or choosing K for K-Means. Given these results, you are expected to report the best configuration. Note that creativity of your trials is always appreciated and will be reflected to you as bonus!

One little detail. The only libraries you are allowed to use are *numpy* and *rouge*[2] apart from Python's standard libraries. So, you will implement K-Means and cross-validation mechanisms by yourselves. Yet, for side operations, such as plotting, you can use any library.

[1] For example, you can consider a fixed K, K as a function of document length, or an automated elbow method.
[2] `https://pypi.org/project/rouge/`

3 K-fold Cross-Validation

Though this part is also covered in class, it might be better to explain it in detail to prevent ambiguity. K-fold cross-validation is a mechanism that is mostly used to find hyper-parameters of a model that would yield the best performance when it is in production. Thus, it can be used to compare different models quantitatively. To do so, one can follow the steps below:

1. Split the dataset into two as *train* and *test* splits and do not conduct *any* operation on the test split, including preprocessing, tf-idf vectorization, dimensionality reduction or dictionary construction.
2. Construct some models, let's say *M1* and *M2*.
3. Split the train split to k-folds.
4. Train M1 and M2 on k-1 fold and compute its score in the remaining fold. Treating each fold as the test fold only once, repeat this process k times.
5. Compute mean and standard deviation of the scores on the previous step for both models separately. Call this statistic as *validation score.*
6. The model with higher *validation scores* is the superior one. Train it on whole dataset and compute its[3] score on the test split you have spared before. Note that pretraining operations such as text normalization, tf-idf vectorization, dimeansionality reduction etc. must be identical in the training and test split. So, the coefficients or parameters of these operations must be learned from the training split.

Thus, as the result of the cross-validation, you are expected to generate a table similar to Table 1. Based on these scores, in the conclusion, comment on the ups and downs of your models and why one has outperformed others. Note that you are encouraged to support your deductions with necessary statistics, plots and figures.

Model	**Metric**	**Validation Score**	**Test Score**
Model 1	*Rouge 1*	$\mu_{11} \pm \delta_{11}$	ts_{11}
	Rouge 2	$\mu_{12} \pm \delta_{12}$	ts_{12}
	Rouge L	$\mu_{1L} \pm \delta_{1L}$	ts_{1L}
Model 2	*Rouge 1*	$\mu_{21} \pm \delta_{21}$	ts_{21}
	Rouge 2	$\mu_{22} \pm \delta_{22}$	ts_{22}
	Rouge L	$\mu_{2L} \pm \delta_{2L}$	ts_{2L}

Table 1: Cross-validation Results

4 Submission

Here are a couple of notes on submission:

- You are expected to submit an IPython Notebook that is runnable. For utility functions such as text preprocessing, file loading etc., you may use an external script as well to keep the notebook less crowded. Yet, it is totally fine to submit a single notebook with all the codes inside.
- You are already provided with a notebook template that reflects how the project flow might be. This is just to ease grading process and you are totally free to edit/remove/insert any piece of code as long as resulting notebook is easy to follow.

[3]For the sake of this project, you are expected to compute all models' test scores.

- During evaluation your notebook will be run from scratch. Please leave a note on each cell about how many minutes it takes to run it, so that I know what will face. If any of the computations takes longer than 10 minutes, you can upload the result of that computation as a pickle and submit it as well. This way, I can run your notebook without waiting.
- Your code will be placed one level below the dataset and GloVe pickle. For ease of grading, please use following paths:
  - `VECTOR_PATH = "../glove.6B.200d.pkl"`
  - `ARTICLES_PATH = "../articles/"`
  - `GOLD_SUMMARIES_PATH = "../gold_summaries/"`
- Lastly, name your notebook with your student id (e.g. 2018700100.ipynb) and ***do not write your name inside the notebook or anywhere else*** for the sake of blind review.
- Late submission policy is the same with previous assignments.

5 Grading

Here is the tentative grading scheme:

1. **Submitting a runnable .ipynb: 10 pts**
2. **Sentence Representation: 20 pts**
3. **K-Means Clustering: 25 pts**
4. **Model Selection: 25 pts**
5. **Conclusion: 10 pts**
6. **Notebook Organization: 10 pts**
7. **Bonus: 10 pts**