ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν Πανεπιστήμιον Αθηνών
ΙΔΡΥΘΕΝ ΤΟ 1837

ATHENA Research & Innovation Information Technologies

M915 - NLG & NLU

Extractive Summarizer

Report

Christina-Theano (Theatina) Kylafi

LT1200012

Tasks

The main task of the project comprises two subtasks:

1. Sentence Scoring
2. Extractive Single-Document Summarization (using task 1)

Data Processing

The steps followed in the pre-processing stage are the following:

1. Kept only the extractive summary types of the dataset ()
2.
3.

Kept only the extractive summaries

Split train data in 4 parts due to computational constraints

Labels -> RougeL (with respect to gold summaries based on [SummaRuNNer: a recurrent neural network based sequence model for extractive summarization of documents](https://arxiv.org/abs/1611.04230))

Chosen stats – RougeL / chosen or not

Sentence cleaning before tokenization to create the features

Features (based on Source: [Extractive Summarization using Deep Learning](https://arxiv.org/pdf/1708.04439v1))

Experiments

Several regression algorithms were tested and fine tuned on task 1 (sentence scoring) :

1. ElasticNet
2. LinearSVR
3. SGDRegressor
4. Lasso
5. LassoLars,

with the most appropriate turning out to be no. 3, **SGDRegressor**. Also, scalers were added in a pipeline before the regression models to test their effectiveness on the final performance, concluding that no scaling
After the fine tuning using the dev dataset, the output scores were the following :

10 Fold CV (mean scores)	Train 1	Train 1 + Train 2
MAE (mean absolute error)	0.070	0.069
MSE (mean squared error)	0.016	0.016
R2 (r - squared)	0.155	0.156

A slightly better performance is observed in case of training the best fine-tuned model with both training parts 1 & 2 (smaller error, greater R2).

Training

Train1

Train1+2

Evaluation

Stats and figures

Notes

1.
2.
3.
4.

Notes

1. scoring -> label rougeL -> sentence feats -> train

2. grid search ?! (or manual fine tuning)

3. test -> input doc -> predict score -> keep N first sentences or keep those over a threshold -> create summary -> calculate rouge1/2/L

2. [Text Summarization References](https://github.com/Tian312/awesome-text-summarization/blob/master/README.md)