ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ⸭ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν
Πανεπιστήμιον Αθηνών
ΙΔΡΥΘΕΝ ΤΟ 1837

dit

ΙΕΛ
ILSP

ATHENA
Research & Innovation
Information Technologies

M915 - NLG & NLU

Extractive Summarizer

Report

Christina-Theano (Theatina) Kylafi

LT1200012

Introduction

The main task of the project comprises two **subtasks**:

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

ROUGE-N measures the number of matching '**n-grams**' between the predicted and the golden summary. The **recall** counts the number of overlapping n-grams found in both the model output and reference divided by the total number of n-grams in the reference. **Precision** is calculated in almost the exact same way, but rather than dividing by the **reference** n-gram count, we divide by the **model** n-gram count. ROUGE F1 score is the harmonic mean of the two, that gives us a reliable measure of the summarizer's performance that relies not only on the model **capturing** as many **words** as possible (recall) but doing so **without** outputting **irrelevant** words (precision).

ROUGE-L measures the **longest common subsequence** (LCS) between the summarizer's output and the golden summary. A longer shared sequence would indicate **more similarity** between the two sequences. Recall and precision calculations are applied just like in the case of ROUGE-N score above.

Data Exploration

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

1. Only the **extractive** summary types of the NEWSROOM dataset were kept (dropped abstractive & mixed)
2. **Training** set was split in **4 subsets** due to computational constraints (quicker training)
3. Sentences in the dataset were scored to create the target variable (**labels**), by calculating each one's **RougeL fmeasure** score with respect to the document summaries (based on paper "SummaRuNNer: a recurrent neural network based sequence model for extractive summarization of documents")

Additionally, **statistics** were produced, grouped by whether each sentence was part of the summary(chosen→1) or not (chosen→0).

RougeL **scores**	Chosen == **0** (**mean**)	Chosen == **1** (**mean**)	Chosen == **0** (**std**)	Chosen == **1** (**std**)
Train1 set	0.11	**0.35**	0.12	0.24
Train2 set	0.11	**0.58**	0.13	0.29
Test set	0.11	**0.46**	0.13	0.30
Dev set	0.11	**0.46**	0.13	0.30

The table above confirms the good fit of RougeL fmeasure for sentence scoring, as **high RougeL** values most likely indicate the sentence's **participation** in the respective **summary**.

Then, **feature engineering** was applied:

1. **Text cleaning** methods were performed to the sentences for feature extraction
2. **Feature extraction** methods were applied on the data, based on paper "Extractive Summarization using Deep Learning", creating **5 feature** variables:
   i. **thematic words** ratio → ratio of sentence thematic words (**10** most **frequent** words in the text) to total sentence words
   ii. sentence **position** → high feature value towards the beginning and ending of the document, and a progressively decremented value towards the middle
   iii. sentence **length** → excludes sentences that are too short (threshold=3)
   iv. **paragraph related** sentence **position** → focus on the start and end of each paragraph
   v. **numerals** ratio → ratio of figures (numerical data) to total sentence words

Training

After the feature extraction, the **sentence scorer** is **fine tuned** (grid search technique) using **dev** set, deploying multiple regression models and the best model candidate is trained using **train** set. Finally, as it is mentioned in section "**Evaluation**" below, the scorer is employed to score document sentences so as to create the respective extractive summaries (summarizer) and evaluation is performed using **test** set (utterly unseen data).

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** as follows :

- alpha = **8.192e-10**
- max_iter = **1000**
- tol = **6.4e-5**
- epsilon = **3.2e-4**
- learning_rate = "**adaptive**"
- loss = "**squared_error**"
- penalty = "**elasticnet**"

Also, **scalers** (standard, min-max, max-abs) were added **externally** in a pipeline before the regression models to test their effectiveness on the final performance, concluding that **internal** scaling (**penalty** / **alpha** parameter fine-tuning) results in **better** model performance.

After the final regressor's fine-tuning, the output test **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).

Evaluation

After the training and evaluation of the **sentence scorer**, the regression model is employed as part of the **summarizer**, to calculate scores for the document sentences. Then, the first **N** sentences are chosen (highest RougeL scores), on condition that except the 1st one, the other all have a score over a specific threshold **thr**.

Below, the Rouge scores (**1** / **2** / **L**) of multiple N - threshold experiments are presented :

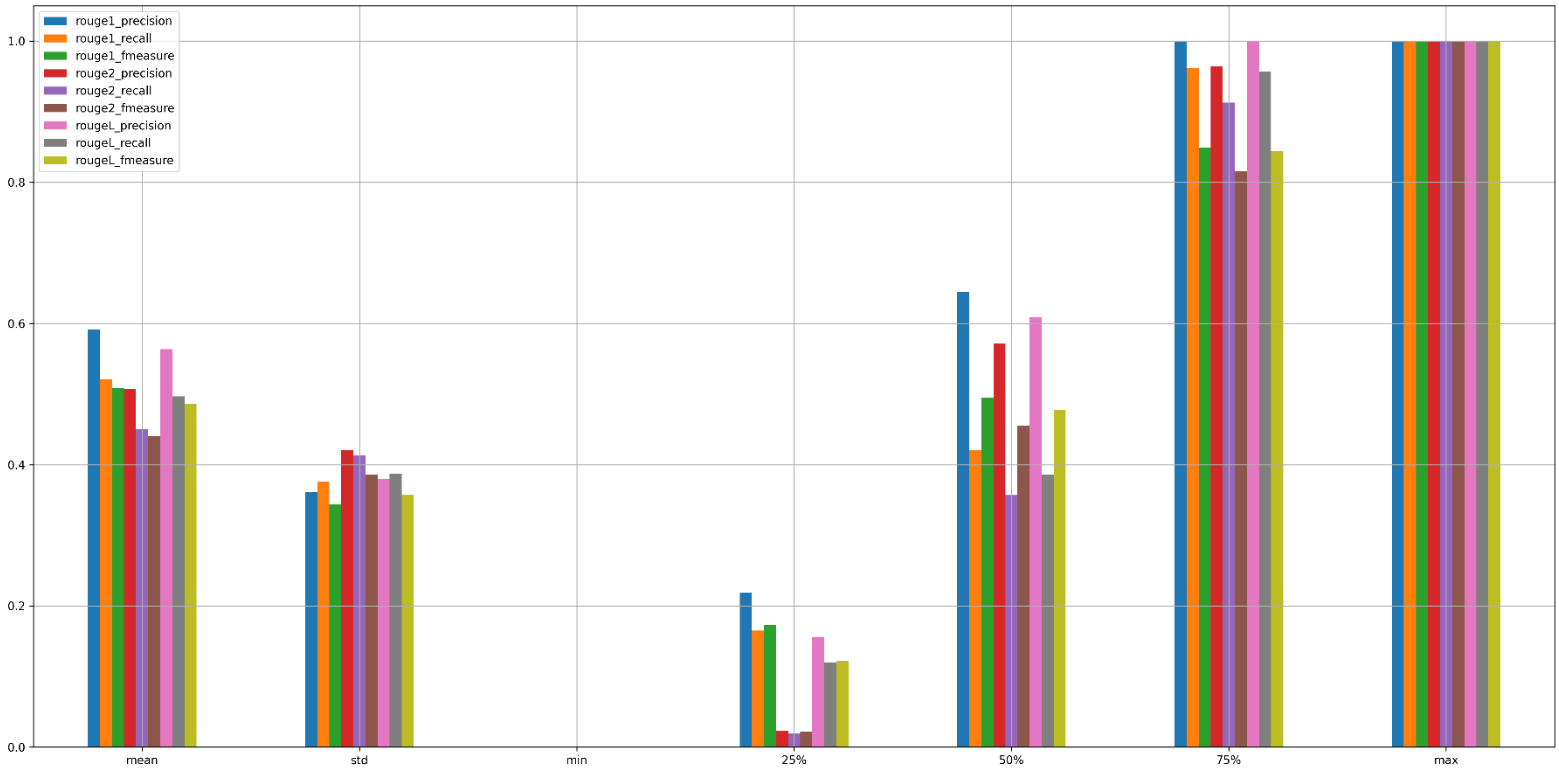
Rouge	Rouge1	Rouge1	**Rouge1**	Rouge2	Rouge2	**Rouge2**	RougeL	RougeL	**RougeL**
Mean Scores	(P)	(R)	**(F)**	(P)	(R)	**(F)**	(P)	(R)	**(F)**
N=**1**, thr= ---	0.59	0.52	**0.51**	0.51	0.44	**0.44**	0.56	0.49	**0.49**
N=**2**, thr=**0.3**	0.54	0.58	0.50	0.46	0.50	0.43	0.51	0.55	0.48
N=**2**, thr=**0.4**	0.59	0.52	**0.51**	0.51	0.45	**0.44**	0.56	0.49	**0.49**
N=**3**, thr=**0.2**	0.44	0.62	0.47	0.37	0.53	0.40	0.41	0.58	0.44
N=**3**, thr=**0.3**	0.54	0.58	0.50	0.46	0.50	0.43	0.51	0.55	0.48
N=**3**, thr=**0.35**	0.59	0.52	**0.51**	0.51	0.45	**0.44**	0.56	**0.50**	**0.49**
N=**3**, thr=**0.5**	0.59	0.52	**0.51**	0.51	0.45	**0.44**	0.56	0.49	**0.49**
N=**4**, thr=**0.2**	0.44	0.62	0.47	0.37	0.53	0.39	0.41	0.58	0.44
N=**4**, thr=**0.4**	0.59	0.52	**0.51**	0.51	0.45	**0.44**	0.56	0.49	**0.49**
N=**5**, thr=**0.1**	0.22	0.80	0.32	0.18	0.67	0.26	0.20	0.74	0.29
N=**5**, thr=**0.25**	0.46	0.61	0.48	0.39	0.52	0.41	0.43	0.57	0.45
N=**5**, thr=**0.3**	0.54	0.58	0.50	0.46	0.50	0.43	0.51	0.55	0.48
N=**5**, thr=**0.35**	0.59	0.52	**0.51**	0.51	0.45	**0.44**	0.56	**0.50**	**0.49**
N=**5**, thr=**0.4**	0.59	0.52	**0.51**	0.51	0.45	**0.44**	0.56	**0.50**	**0.49**
N=**5**, thr=**0.45**	0.59	0.52	**0.51**	0.51	0.45	**0.44**	0.56	0.49	**0.49**
N=**5**, thr=**0.5**	0.59	0.52	**0.51**	0.51	0.45	**0.44**	0.56	0.49	**0.49**
N=**8**, thr=**0.3**	0.54	0.58	0.50	0.46	0.50	0.43	0.51	0.55	0.48
N=**8**, thr=**0.5**	0.59	0.52	**0.51**	0.51	0.45	**0.44**	0.56	0.49	**0.49**

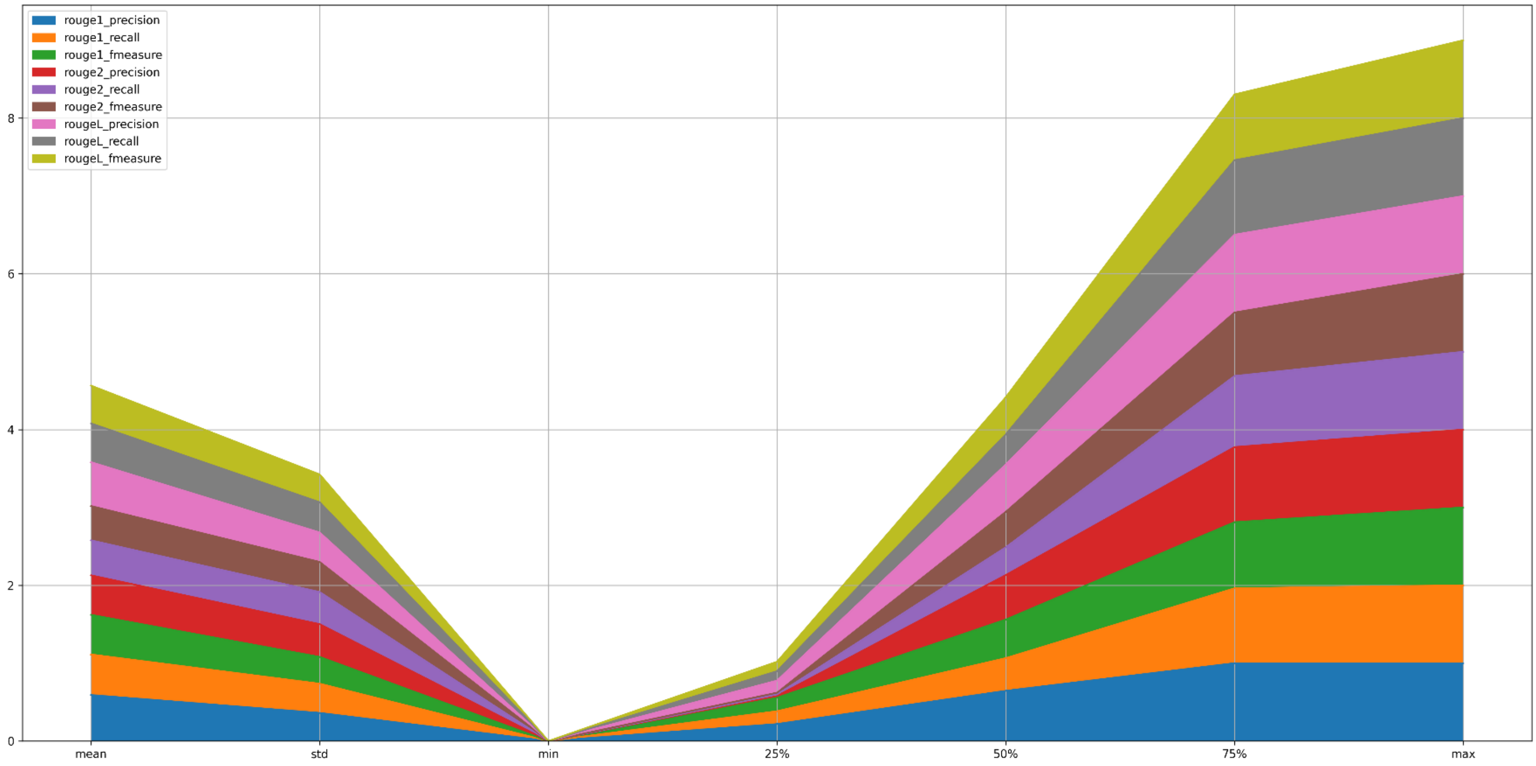
The scores above were rounded to two decimal numbers, in order to acquire a wider viewpoint of the N - threshold combination results (minor score differences might be of higher importance in future work) .

The score table indicates that even N=1 sentence could constitute a good summary (depending on the document's length), as well as for N>1 on condition that configurations are made to the threshold value.

In general, the higher the threshold values the higher the final Rouge scores, possibly due to the elimination of multiple low scored sentences (irrelevant / redundant) which results in a decreased total of sentences thus forming a summary exclusively consisting of high scored, informative sentences (purpose of the main task).

The scores of combination N=5 , threshold=0.35 are depicted in the colourful figures below :


rouge1_precision
rouge1_recall
rouge1_fmeasure
rouge2_precision
rouge2_recall
rouge2_fmeasure
rougeL_precision
rougeL_recall
rougeL_fmeasure
mean
std
min
max


rouge1_precision
rouge1_recall
rouge1_fmeasure
rouge2_precision
rouge2_recall
rouge2_fmeasure
rougeL_precision
rougeL_recall
rougeL_fmeasure
mean
std
min
max