Extractive Summarization 2

Bertelli Davide (223718)
University of Trento
Via Sommarive, 9, 38123 Povo,Trento TN
davide.bertelli-1@studenti.unitn.it

1 Abstract

The final project "Extractive summmarization 2" aims at experimenting with scoring strategies developed to rank sentences and synthetize a summary. To perform this task, features like the term frequencies of the tokens in the sentences are used to find the best summarization. The evaluation of the generated summary is done through the Rouge-N score, specifically the Rouge-2. Two optimisation tasks are used to achieve the best possible results. During the computations the *spacy* library has been used being highly efficient for the processing steps. In this work the summarization performance of different sets of metrics are investigated. The best results are achieved exploiting the lemmatized version of the sentences.

2 Framework Structure

To achieve a summary of the documents, a custom data structure has been built in order to better handle the available data. The framework was divided in different classes:

- The **Scores Class** is the one responsible to store the results of the available scoring strategies applied to the sentences of the documents.
- The **Sentence Class** is the one responsible to store any sentence of the documents. It contains the identifier of the sentence and calls the specific methods of the Scores class to get a ranking of itself.
- The **Document Class** is the one responsible to store each document provided in the input. In it each sentence is stored in a Sentence class instance saved into a specific dictionary in which the sentences are addressed by a string containing the document identifier followed by the sentences position in the document.
- The **Dataset Class** is the one responsible for storing the available documents. It stores the input documents in a specific dictionary in which each of them is addressed by their identifier. Some document features are saved together with the documents in order to decrease the amount of memory required by the data structure and enforce the sharing of knowledge across the dataset. Specifically, the features saved are the tokens identified as numerical, the named entities found and the document frequencies of the tokens.

3 Data Available

The dataset used in this project is the *CNN_dailymail 3.0.0* [1] which is a collection of newspaper articles about daily news written by journalists employed at CNN and at Daily Mail. The dataset has over 300 thousands entries divided in training and test sets. To perform the experiments described in section 5, only the training part was used. Each entry of the dataset is organised as a dictionary with three keys:

- **id**: It is the unique identifier of each article.
- **article**: It contains the text of the article written by the journalists. The information of the font used or other font modifiers is lost.
- **highlights**: It contains the summary of the article. It is written by humans and for this is a "gold summary" useful also for an abstractive summarization task.

3.1 Processing Steps

The input data were processed with the framework defined above, by performing the following steps. Firstly, the *Spacy*[3] pipeline for the english language was loaded, *en_core_web_md*, followed by the *textrank* pipeline coming from the python module *pytextrank*[2] to perform the sentence ranking task. The medium, *md*, model of *spacy* was used because it is large enough to support the sentence similarity computation. Then all sentences, in each document, went through the pipeline which outputs an object comprehensive of the tokens they are made of and their rankings. At this point, the processed documents were stored into a dictionary of Document items where each of them is addressed by its ID. The reference summary was stored, without processing, directly into the Document class of the document it belongs to. Each processed sentence of the article was also stored in the Document class, inside a dictionary of Sentence items addressed by their Sentence ID: a string containing the Document ID and the sentence location. Then each token in the processed sentences underwent the term frequency computation and it was checked and saved into a specific set in the Dataset class according to their type: named entity, proper noun or numerical token. It was chosen to save those tokens in the highest level of the class hierarchy both to reduce the amount of physical memory required by the structure and also to enforce the knowledge sharing among documents. The text ranking performed by the spacy pipeline was instead saved into a dictionary belonging to the Document class, where each value is addressed by the sentence ID of the sentences used in the similarity computation. The sentence similarity was also computed and saved in a dictionary in which the scores are addressesed by a string made up of the Sentence IDs of the sentences used in the computation. When the processed sentences are passed to the Document class and the Sentence class instance is built, this latter initiates all the scoring methods to 0. At the end of the building process above, if requested, the

scores computation of the sentences takes place and the data structure is now able to provide a summary.

4 Metrics

The metrics in this work derived from *Meena and Gopalani*[4], but were extendended to include also the Term Frequencies computation, Similarity score, Numerical score and Thematic words scoring methods. All the scoring strategies refer to each sentence in the documents. The following is a comprehensive lists of the metrics here used:

1. Term Frequency: The score is computed by summing up the term frequency of each token in the sentence. The term frequency is the number of times a token is found in the document divided by the total number of tokens contained by the document.

2. Sentence Location: The score is computed using one of the scoring strategies among the one introduced by *Edmundson*[5] in which a sentence is given score 1 if it is one of the first two of the document, the three scores implemented in the paper of *Nobata et al.*[6], and the one proposed in the paper of *Fattah and Ren*[7]. The selection is done using a vector of flags. Furthermore, it is possible to combine all of them and set the score to be the maximum among the results.

$$S_{ED_{loc}} = \begin{cases} 1 & S_{pos} \leq 2 \\ 0 & \text{otherwise} \end{cases}$$

$$S_{NB1_{loc}} = \begin{cases} 1 & S_{pos} < threshold \\ 0 & \text{otherwise} \end{cases}$$

$$S_{NB2_{loc}} = \frac{1}{S_{pos}}$$

$$S_{NB3_{loc}} = max(\frac{1}{S_{pos}}, \frac{1}{S_{len} + S_{pos} + 1})$$

$$S_{FaR_{loc}} = \begin{cases} 1 - (S_{pos}/5) & S_{pos} < 5 \\ 0 & \text{otherwise} \end{cases}$$

Where S_{pos} and S_{len} are, respectively, the position of the sentence in the document and its length.

3. Proper Noun: The score is computed using the method provided by *Nobata et al. 2001*[6]. The score is the sum of the term frequencies for those tokens in the sentence being identified as proper nouns.

4. Co Occurrence: The score is computed by summing up all the term frequencies for those tokens belonging to the summary and to the sentence.

5. Similarity Score: The scoring strategy was proposed by *Fattah & Ren 2009*[7]. Its computation is performed using the similarity method embedded in the objects of the *spacy* module. For each document the similarity score is computed among all the possible couples of sentences.

6. Numerical Score: The score is computed using the method introduced by *Fattah & Ren 2009*[7]. It is computed as the number of tokens identified as numerical ones, divided by the number of tokens in the sentence, i.e the sentence length.

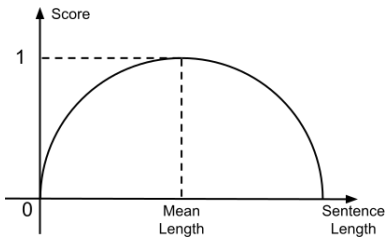
7. TF-IDF: This score is computed following the method introduced by *Nobata et al. 2001*[6].

$$TF - IDF = \frac{tf}{1 + tf} \log \frac{DN}{DF}$$

Where tf is the term frequency of the token, DN is the number of document available and DF is the document frequency of the token.

8. Sentence Rank: The score is computed using a ranking algorithm implemented by Mihalcea et al. [8]. For its computation a specific *spacy* pipeline is employed[2].

9. Sentence Length: This score is computed using the length of the sentence. Given the lack of indications in the reference paper [4], a custom method was implemented. The score is computed as the result of a parabola function as stated below.



$$S_L = \frac{-1}{S_{ml}} S_l^2 + \frac{2}{S_{ml}} S_l$$

With S_L being the sentence length score, S_{ml} the mean length of the sentences in the document and S_l the length of the sentence under evaluation.

10. Positive and Negative: The scores are computed by checking the frequency of the tokens with respect to the sentences in the summary. If the tokens of the sentence are highly frequent in the summary they are said to be positive, otherwise negative, as shown by *Fattah & Ren*[7].

11. Thematic Words: The score is computed by checking if the tokens in the sentence have an high frequency. If so, they are said to be thematic and their frequency is added to the score. To chose whether the frequency is high enough, a threshold of two times the mean frequency of the tokens was used.

12. Named Entities: The score is computed by summing up the frequencies for those tokens identified as named entities.

5 Experiments

In this section the results proposed are derived using a set of 100 documents taken sequentially from the *CNN_dailymail* dataset. This represents a first difference with respect to the reference paper in which only 6 documents were exploited. An analysis of the effect in increasing the number of documents is provided in subsection 5.2. In evaluating the results the Rouge-N score is used, specifically the Rouge-2 is employed being more conservative than the Rouge-1.

5.1 Meena and Gopalani strategy

The first experiment consisted in exploiting the same combinations of scoring strategies that were used in the ref-

erence paper[4] and apply them to the *CNN_dailymail* dataset. This was done to understand how much the Rouge-2 evaluation diverges from the results obtained in the paper. Of note, while *Meena and Gopalani* used the *DUC 2002 dataset*, here it is used the *CNN-dailymail*. For clarity, the combinations of the scoring strategies investigated by Meena and Gopalani in their work are shown in table 1.

Comb1	TF-IDF, Word Co-occurrence, Sentence Length
Comb2	Word Co-occurrence, Sentence Length, Sentence Location
Comb3	TF-IDF, Word Co-occurrence, Sentence Length, Sentence Location
Comb4	Sentence Length, Sentence Location, Named Entity, Positive Keywords, Proper Noun
Comb5	Word Co-occurrence, Sentence Length, Sentence Location, Named Entity, Positive Keywords, Proper Noun
Comb6	TF-IDF, Word Co-occurrence, Sentence Length, Sentence Location, Named Entity, Positive Keywords, Negative Keywords, Sentence Rank
Comb7	TF-IDF, Word Co-occurrence, Sentence Length, Sentence Location, Named Entity, Positive Keywords, Negative Keywords

Table 1: *Combinations of scoring strategies used in the reference paper.*

The results obtained by the computations for each combinations are shown in table 2. The data were collected using a dataset of 100 documents and represent the mean over the dataset for each measure.

Combination	Rouge-2	Precision	F1-Score
Comb1	0.1146	0.0356	0.0538
Comb2	0.0703	0.0481	0.0553
Comb3	0.1176	0.0369	0.0556
Comb4	0.1469	0.0760	0.0975
Comb5	0.1443	0.0744	0.0956
Comb6	0.1200	0.0375	0.0565
Comb7	0.1191	0.0373	0.0561

Table 2: *Results of using the combinations of Table 1 over the CNN_dailymail dataset*

As shown in table 2, the best combination which leads to the highest values for Rouge-2, Precision and F1-score is the combination 4, as concluded by the reference paper.

5.2 All available scores

To understand the overall performance of the scoring strategies available, all of them were combined to synthesize the summary of a set of documents belonging to the CNN dataset, and the average of the Rouge-2 measurements was taken. The model was tested using 6 documents, as in the reference paper, 100 and 1000 documents. The results are shown in table 3. Seen the small improvement in using 100 or 1000 documents, the following experiments has been conducted using 100 documents. We can observe how over 6 documents we have a lower Rouge-2 score due to high variation in the summary results and a lower knowledge of the types of tokens in the dataset. Furthermore, using all the scoring strategies led to lower results due to the Negative keywords score which adds a negative score which increases for long sentences, like the ones in news articles, even if a normalization factor is used.

#Documents	Rouge-2	Precision	F1-score
6	0.1094	0.0290	0.0454
100	0.1202	0.0374	0.0564
1000	0.1232	0.0382	0.0577

Table 3: *Results of applying the scoring strategies available over different size datasets*

5.2.1 Sentence Location Score analysis

Usage of all the location scores: Instead of using just one of the available Sentence Location scoring methods, an idea was to combine them all and set the Sentence Location score to the maximum among them. The idea came from the desire to achieve a maximum Rouge-N score and using all the sentence location scores allows to assign the highest score possible to each sentence in the document, lowering the probability that the best sentence for the summary is not considered. This results into a slight improvement of the overall scores: 0.1221 for Rouge-2, 0.0386 for Precision and 0.0579 for F1-score.

Nobata et al. Sentence Location 1: In the paper of Nobata et al.[6], the researchers introduced a sentence scoring method which exploits the sentence location and compares it with a threshold. The nature of the threshold was not indicated in the paper and so its influence on the results was investigated. Table 4 shows the results obtained varying the threshold in the range 1-20 and considering the highest value among the scoring strategies used.

Treshold	Rouge-2	Precision	F1-score
1,2,14,20	∼0.1188	∼0.0369	∼0.0556
3,4,6-8,16-19	∼0.1214	∼0.0379	∼0.0571
5,15	∼0.1223	∼0.0383	∼0.0577
9-13	∼0.1198	∼0.0373	∼0.0563

Table 4: *Results of Rouge-2 varying with the sentence location threshold, using the sum of all the sentence location method.*

As we can see, some subsets of values for the threshold, result into a slight improvement of the Rouge-2, even if the Precision and F1-score were not particularly influenced. Applying the same algorithm, but using only *Nobata et al.* threshold scoring method, the results in table 5 were obtained.

Treshold	Rouge-2	Precision	F1-score
1, 9-14, 20	∼0.1194	∼0.0371	∼0.0560
2-8, 16-19	∼0.1214	∼0.0379	∼0.0571
15	∼0.1225	∼0.0380	∼0.0573

Table 5: *Rouge-2 results at varying Nobata sentence location score and using it during the computations*

5.3 Lemma Usage

When writing a summary, humans often take words from the article and reuse them in a novel grammatical structure. This may interfere with the Rouge-2 and the scoring

methods computations. To circumvent this, it was though to rely on the lemma of each token.

Setup	Rouge-2	Precision	F1-Score
Without Lemma	0.1221	0.0386	0.0579
With Lemma	0.1561	0.0597	0.0854

Table 6: *Results of Rouge-2 scores in using or not the Lemma representation.*

As observed in table 6, all the available metrics improved, likely thanks to the the general essence of the lemma which may have positively influenced the scoring methods and so the synthetized summary.

5.4 Weights Introduction

In the reference paper the researchers expressed the will to introduce, in the future, a set of values to make a weighted sum of the scoring strategies and thus improve the results. This latter feature was added in this work to understand if it existed a linear relation among the scoring strategies able to improve the results. These weights are applied only during the summary computation. To chose their value, an optimisation task was run, using the *optuna*[9] module, and each weight was optimised over an interval from -10 to +10 with a step size of 0.5. The algorithm run 500 trials before stopping and the parameter to optimize was the Precision, due to its ability to idicate the overal usefullness of the summary. The weights found in the optimisation task are shown in table 7. The rouge-2 score obtained introducing the optimized weights are illustrated in table 8. Specifically, weights higher and equal than 9 lead to the best results for Rouge-2.

Scores	Weight	
	Without Lemma	Lemma
Term Frequency	-3.5	1.5
Sentence Location	1.5	0.0
Proper Noun	-8.5	0.5
Word Co-occurrence	-0.5	-6.5
Sentence Similarity	0.5	-0.5
Numerical Tokens	-4.5	4.0
TF-IDF	0.0	0.0
Sentence Rank	1.5	-2.5
Sentence Length	-1	0.0
Positive Keywords	7.5	3.5
Negative Keywords	10.0	9.0
Thematic Words	-7.5	-1.0
Named Entities	-4.5	7.0

Table 7: *Weights result after the optimization task.*

Weights	Rouge-2	Precision	F1-score
Without Lemma	0.1951	0.1452	0.1564
Lemma	0.2570	0.2263	0.2278

Weights ≥ 9	Rouge-2	Precision	F1-score
Without Lemma	0.2148	0.1436	0.1635
Lemma	0.2592	0.2180	0.2254

Table 8: *Rouge-2 after applying the weights in the summarization.*

5.4.1 Best representing subset

Given the results obtained in table 7 and 8, I decided to setup an optimisation task to identify which subset of scoring strategies would have performed better on the dataset. The optimization was done on the Recall, Precision and F1-score metrics, its results are shown in table 9 and 10.

Metric optimized	Rouge-2	Precision	F1-score
Rouge-2	0.1311	0.0840	0.0948
Precision	0.1762	0.0975	0.1237
F1-Score	0.1171	0.0361	0.0546
With Lemma			
Rouge-2	0.1354	0.0572	0.0795
Precision	0.2591	0.2179	0.2254
F1-Score	0.1138	0.0623	0.0792

Table 9: *Rouge-2 results depending which metric has been optimized.*

Metric optimized	Best subset
Rouge-2	Sentence Similarity, Sentence Length, Sentence Location, Proper Nouns, TF-ISF-IDF, Word Co-Occurrence
Precision	Negative Keywords
F1-Score	Thematic Features, Sentence Length, Proper Nouns, Sentence Rank, Numerical Tokens

Table 10: *Best representing subsets using the lemma and their Rouge-2 scores.*

6 Conclusions

Comparing the results of this work with those from *Meena and Gopalani* [4], we can observe that both analysis highlight combination 4 as the best set of scoring strategies for the summarization. Notwithstanding, the results presented in their paper showed a Rouge-2 higher than the one in my experiments. This may be due to the smaller dataset used in the reference paper or to an algorithm highly fine-tuned. Furthermore, the diverse nature of the dataset may explain the difference. Indeed, the *CNN_dailymail*, here used, is built from newspaper articles which are relatively long in size, and may also embed some redundancies in the informations they contain; this may be a positive aspect during score computation, but it highly increases the probability that the "best" sentence for the summary is masked by other similar ones. The lemma usage provided the best results due to its capability of projecting a text with all its grammatical enrichments into a general representation space allowing the model to partially overcome the gap between the summary written by humans and the artificially generated one. Lastly, the introduction of optimized weights showed that it exists a subset of scores providing a better summarization; however the scores composition of this subset depends on the nature of the document to summarize (gossip, sport, etc.) and on the usage of a general representation space like lemmas. Indeed, scores discouraged in a lemma free space (e.g., Named Entities score) had been highly increased when the lemma was used to represent the documents.

References

[1] https://huggingface.co/datasets/cnn_dailymail

[2] https://spacy.io/universe/project/spacy-pytextrank

[3] https://spacy.io/

[4] Y. K. Meena and D. Gopalani. 2014. *Analysis of Sentence Scoring Methods for Extractive Automatic Text Summarization.* In *Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies* (ICTCS '14). Association for Computing Machinery, New York, NY, USA, Article 53, 1–6. DOI: https://doi.org/10.1145/2677855.2677908

[5] H. P. Edmundson. *New methods in automatic extracting.* J. ACM, 16(2):264–285, Apr. 1969.

[6] C. Nobata, S. Sekine, M. Murata, K. Uchimoto, M. Utiyama, and H. Isahara. *Sentence extraction system assembling multiple evidence.* In *In Proceedings of the Second NTCIR Workshop Meeting*, pages 5–213, 2001.

[7] M. A. Fattah and F. Ren. *Ga, mr, ffnn, pnn and gmm based models for automatic text summarization.* Computer Speech and Language, 23(1):126 – 144, 2009.

[8] R. Mihalcea and P. Tarau. *Textrank: Bringing order into texts.* In D. Lin and D. Wu, editors, Proceedings of EMNLP 2004, pages 404–411, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[9] https://optuna.org/