# NLP project: Text Summarization

Apostol Laurentiu
*s317518@studenti.polito.it*

Zerbini Ilaria
*s333246@studenti.polito.it*

***Abstract*—This project is a proof of concept that aims to extend the ”Mind the Gap! Injecting Commonsense Knowledge for Abstractive Dialogue Summarization” [1] paper by introducing a novel approach that tries to enhance text summarization with Natural Language Processing (NLP). Recognizing a possible limitation in the original paper’s method for selecting the most informative elements from a Commonsense Knowledge model (COMET) [2] to inject into a dialogue abstractive summarization task, we propose a different selection mechanism. Instead of basing the selection solely on the similarity scores between input sentences and Commonsense Knowledge assertions, we aim to incorporate task-specific criteria by trying to predict the similarity between golden summaries and said assertions. This method seeks to train a model capable of determining the relevance of COMET-generated assertions to the summarization objective, thereby refining the selection process within the summarization framework.**

**Through comparative analysis and evaluation, we have found that both methods produce summaries of similar quality, fact which appears to indicate a low importance of the COMET selection process. By trying to test this last hypothesis we however discovered that randomly selecting the COMET assertions for each dialogue achieves remarkable performance, surpassing both the previous method and our proposed one.**

## I. Introduction (problem statement)

Text summarization is a key task in the field of Natural Language Processing (NLP), which focuses on condensing large amounts of text into shorter versions that still capture the essential meaning of the document. When dealing with an ever more information-overloaded world, the need to quickly understand large volumes of information without reading through everything in detail has become essential.

There are two main types of text summarization: extractive and abstractive.

Extractive summarization involves identifying and compiling key phrases or sentences from the original text in order to create a summary. It’s like highlighting the most important parts of a document without changing any words. On the other hand, abstractive summarization goes a step further by rewriting the essence of the text in a new, shorter form. This requires a deeper understanding of the text, since the goal is to produce summaries that are not only concise but also fluent and coherent, often introducing new phrases and sentences that were not present in the original text.

Usually, the context in which this task is studied is comprised of informative documents, whereas dialogues prove to be more challenging, since ad hoc models and suppositions must be made.

Dialogue summarization aims to create shorter summaries while preserving the context of a conversation [3]. Unlike conventional document-to-document summarization (e.g. news articles and scientific publications), this sub-branch suffers from a discrepancy between input and output forms, which makes learning their mapping patterns more complicated.

The biggest difference between informative documents and dialogues consists in the necessity, in the case of dialogue summarization, of detecting unspoken intentions. Another issue is retrieving information from sentences with hidden meaning or common sayings.

To address these problems, Commonsense knowledge models [4] such as COMET can be used to generate a set of event-centered (e.g., HINDEREDBY, XREASON, XNEED) and social interaction (e.g., XINTENT, XWANT) commonsense inferences.

It is reasonable to assume that incorporating such inferences will help improving the summarization model’s performance. This application of Commonsense Knowledge models has not been deeply studied and one of the first moves in this direction is the reference paper for this project, in which the SICK, and its extension SICK++, models are presented, with the aim of properly injecting Commonsense Knowledge into state-of-the-art language models (e.g., BART [5]) for abstractive dialogue summarization.

Although adding useful information to a model’s input usually helps its understanding, doing so without any selection criteria can impede the summarization goal, since (a) expanding source contents is a counter-intuitive approach for the goal of condensation, and (b) simply adding additional inputs in pre-trained language models does not lead to robust inferences [6]. In order to overcome these two issues the authors apply (a) filtering and (b) robust training. The Commonsense Knowledge to be injected as additional context of the dialogue inputs is chosen based on the similarity scores between an input sentence and each of its generated Commonsense Knowledge assertions, that is to say that the most similar assertion is chosen.

In SICK++, the authors also design a new auxiliary task, named ”commonsense supervision”, and jointly train the BART model on the previous task and the new one. Using commonsense knowledge generated from golden summaries as additional supervision (the new target), the goal of the newly added task is to generate the target commonsense. This task is added in the hope of forcing the shared encoder to pay attention to the additional commonsense assertions.

### A. Expected input

The original SICK/SICK++ model was trained and tested on two different datasets: SAMSum [3] and DialogSum [7]. The former is a collection of dialogues, made and annotated by linguists, that aims at creating a compendium of online messaging-like dialogues; the latter is a compilation/selection of samples from three different datasets (Dailydialog [8], DREAM [9] and MuTual [10]), with the intent of creating a dataset containing more formal and informative dialogues, in contrast with the more casual like dialogues of SAMSum.
In this project we have focused on only SAMSum, with only 13% of its original training set size (still more than double the validation and test sets). This was done in order to overcome computational limits in several distinct phases of our pipeline. This project is intended to be a proof of concept for a way to expand the pipeline of the original paper.
The original paper's authors also released the following material:

- a file containing each dialogue sentence (for training, validation and testing sets) and the 25 COMET generated commonsense knowledge proposals (from now on "COMET proposals", in short);
- a file containing both the sentences and only the related selected COMET proposals. The selection process is based on the cosine similarity among the SBERT [11] embeddings of each sentence and of the related 25 initial COMET proposals;
- the code for fine-tuning the BART-based architecture, with each selected COMET proposal appended to the end of each sentence;

Crucially, what was missing from the material provided was the code for generating the filtered COMET proposals and a file with the inclusion of the similarity scores.

### B. Addressed task

In the original SICK paper the authors mention, as stated above, that just adding more text (COMET proposals) to the original input text tends to actually degrade performance and also goes against the task of summarizing [1]. It is therefore necessary to filter the candidate data beforehand, in order to try to add only useful information and in small quantities. In order achieve this, the authors proposed a filtering method based on the similarities among the sentence and the 25 derived COMET proposals. This makes it so that the most similar COMET proposal to the original sentence (hence the most relevant to that sentence) is the one chosen among the 25 options. This, however, also selects COMET proposals that only indirectly enhance the outcome of the main task (summarization). A more direct way would be that of somehow selecting the COMET proposal which is more similar to the golden summary, which is our target. The golden summary is, however, obviously missing at inference time, so that would not be possible. It is, though, indeed possible to try training a shallow neural network architecture in order predict the similarity scores of each COMET proposal with the reference golden summary.
This methodology presents two main issues:

- the model might have a high test loss, meaning that the selected COMET might only be partially relevant;
- the selected COMET proposal might not be relevant enough in regards to the sentence it is referring to;

In order to both test and overcome these two limitations and to try to better study the proposed method, we decided not to filter the COMET proposals based on solely the predicted similarities with the golden summary or on the actual similarities with the sentences, but to take a convex combination of the two similarity scores for each COMET proposal, based on a configurable $\lambda$ parameter, with $0 \leq \lambda \leq 1$. This parameter indicates the proportion of the predicted similarities present in the convex combination.
Finally, we will address the fact that SICK++ appears very similar to the method that we are exploring, although it is not. SICK++ trains the BART model to also be able to output a COMET-like proposal similar to one extracted from the golden summary. The reasoning is that this second task would force the shared encoder architecture to pay attention to the COMET proposal inputs during training and inference. It has to be noted, though, that this has no influence on the selection process of the input COMETS. Our proposal, instead, aims at aligning the selected input COMETS proposals with the desired output (golden summaries).

### C. Expected output

The expected outputs of our pipeline are:

- a set of "golden similarities", that is to say the actual similarity scores of each COMET proposal with the golden summary;
- a shallow neural network model that uses the similarities from the previous point as training, in order to predict the similarities in the absence of the golden summary;
- a set of different files for each $\lambda$ value, each containing all the sentences considered in our project and the top COMET proposal based on the similarity calculated as a convex combination of the sentence similarities and predicted similarities with the golden summary;
- all the code from the previous steps publicly available [1]

## II. Methodology

### A. Overview of the NLP pipeline

The pipeline employed in this project is comprised of the following steps:

- reproducing the missing data and code from the original paper. This also includes checking if the selected COMET proposals (the ones with the best similarities) match the ones provided by the original paper's authors;
- calculating the similarities among the golden summary and each COMET proposal related to the sample of that summary;

[1] https://github.com/angrytako/NLP-SICK-filter-extension/

- training a model on the golden summaries-COMET similarities in order to predict said similarities at inference time;
- combining, via convex combination using the $\lambda$ parameter, the similarities of the original paper's COMET similarities and the predicted similarities with the golden summary;
- keeping only the maximum convex combination similarity COMET proposal per sentence;
- training the entire SICK architecture with the new COMET proposals;

The pipeline described above is the main one, and the original focus of our work. For the rest of the experiments we have also tried to:

- select randomly the COMET proposal;
- select the COMET proposal with the minimum similarity score w.r.t. the reference sentence;

### B. Description of each module

In order to compute the similarities, we first transform the sentences, the COMET proposals and the golden summaries into SBERT embeddings. Note that COMET proposals contain two distinct parts: the type (e.g. ”HINDEREDBY”) and the proposal itself. Only the proposal is used for the embeddings. It is also necessary to prepend the speaker (provided as an additional field) of the sentence to each sentence in order to reproduce the original paper's results.

After calculating the embeddings we can get two types of cosine similarity scores, one for each COMET proposal with its reference sentence and one for each COMET proposal with its reference golden summary.

After computing the similarities, we start training a shallow neural network with the goal of predicting the final similarities for each COMET proposal. We believe that a shallow neural network can work in this case since we are leveraging a bigger and better trained model (SBERT) for the inputs.

The model's training is done with a matrix X as input, where the row $X_i = e_{i,1}||e_{i,2}||...e_{i,25}$, $e_{i,j}$ being the embedding of the $j^{th}$ COMET proposal of the $i^{th}$ sentence. The predicted output for each $X_i$ is $Y_i \in \mathbb{R}^{25}$, representing the similarity scores of a sentence's COMET proposals. The model chosen for the task is a classifier composed of 3 fully connected layers, using RELU as an activation function, and the chosen loss function is MSE:

$$LOSS = \sum_{j=1}^{M} \frac{1}{N} \sum_{i=1}^{N} (\hat{Y_{ij}} - Y_{ij})^2$$

where:

- N is the number of different utterances (25 in our case);
- M is the number of samples;
- $\hat{Y_{ij}}$ is the ”true” similarity of the i-th utterance of the j-th sample;
- $Y_{ij}$ is the similarity of the i-th utterance of the j-th sample to its summary computed by the model;

The model was trained for 100000 iterations, with a normalized test loss of 28%.

Finally, we compute the convex combination of the two similarities ($s\_sent_{i,j}$ and $s\_pred\_sum_{i,j}$) calculated for each COMET proposal $c_{i,j}$, based on the value of $\lambda$.

The target similarity score is computed as:

$$s\_target_{i,j} = \lambda * s\_pred\_sum_{i,j} + (1 - \lambda) * s\_sent_{i,j}$$

Based on the target similarity, the $c_{i,j}$ COMET proposal is chosen for the i-th sentence is the $j_{th}$, where:

$$j = argmax_j(s\_target_{i,j})$$

## III. Experiments

The experiment phase has been divided into two steps, since after the first considerations and results, we decided to explore another hypothesis we came up with.

In Table 1 we report the maximum value for each score obtained for the most interesting configurations.

### A. Experimental Design (extension2)

*1) first approach:* We first approached the study of the parameter $\lambda$, introduced in the previous sections, which is the main hyperparameter of interest for the comparison of our model and the one proposed by the authors.

We have analyzed the different outcomes by changing the parameter with a step size of 0.25, in the range of [0,1], in order to test out different loss scenarios, and we have set the number of epochs to 5.

As we can see in the left graph of Fig. 1, unexpectedly, all the configurations led to similar results, and we couldn't find a clear trend in the data.

In the appendix IV-A all the results concerning the various metrics and configurations are reported.

The results obtained also have overall low values for all ROUGE scores. This is probably just due to the low number of epochs we have set because of computational constraints, compared to the authors' 25, an amount significantly higher.

Moreover, it must be noted that our model did not use all of the original training data (we have used only about 13% of it), and hence, when facing data with high variability, in regards to both topic and language, such as dialogues, this could have damaged significantly the capabilities of the model.

In addition, we cannot select the best value for the parameter $\lambda$, since there we do not observe a stable trend for any of the configurations.

We can only observe that, for the experiments and the setting we have used, there is not a real ”preference” or best choice between the two criteria tested, or among any in-between.

*2) second approach:* Although we did not achieve a conclusive result, this allowed us to explore a different perspective: if the two selection processes lead to a similar results, could this mean that the selection criteria did not matter, as long as

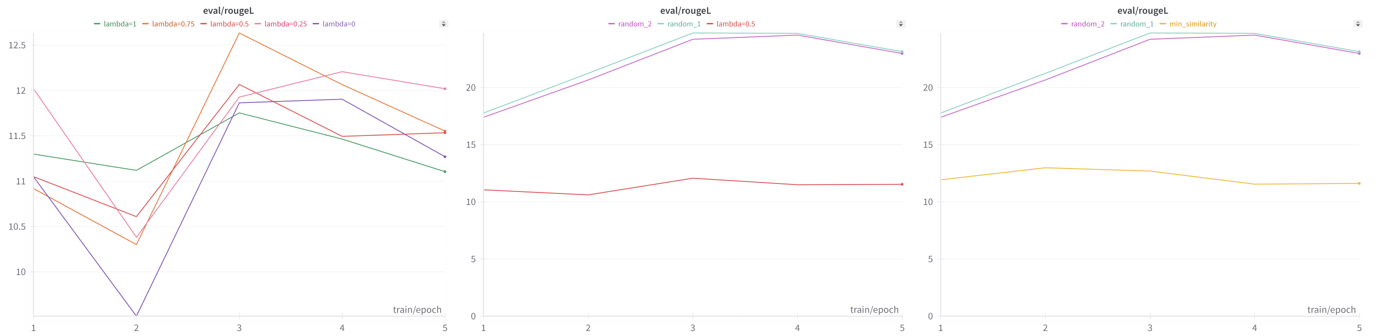
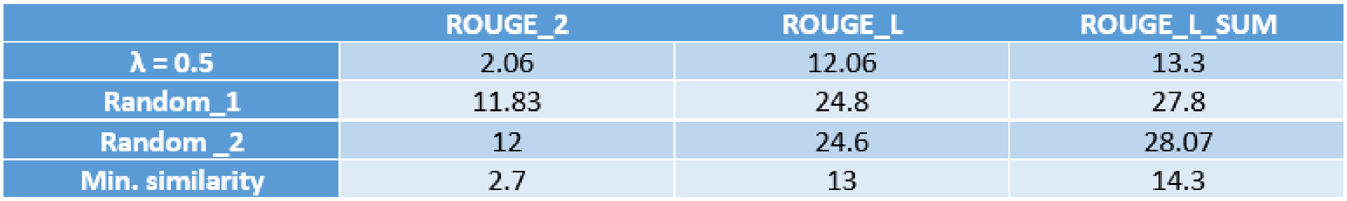


Figure 1: ROUGE-L comparisons among the different experiments. The value for each score is expressed as percentages.



| | ROUGE_2 | ROUGE_L | ROUGE_L_SUM |
|---|---|---|---|
| λ = 0.5 | 2.06 | 12.06 | 13.3 |
| Random_1 | 11.83 | 24.8 | 27.8 |
| Random _2 | 12 | 24.6 | 28.07 |
| Min. similarity | 2.7 | 13 | 14.3 |

Table 1: Rouge best scores comparison; $\lambda = 0.5$, the minimum similarity and the two random experiments. The values for each score is expressed as percentages.

something was selected?

Following this reasoning, we explore a scenario in which the COMET proposal selected for each sentence is chosen randomly. In order to ensure that the results are not a coincidence, two different runs with different seeds are done for the random choice experiments.

In the center graph of Fig. 1 we can see the results from this two runs compared with $\lambda = 0.5$ for reference. The set up used is the same, but the results are significantly better (about three times better) with the randomly selected COMET proposals. This could can be ascribed to two causes: either choosing less ”similar” COMET proposals adds more information (given that each proposal is more different w.r.t. the original sentence, but still relevant since it is derived from it) or adding more varied COMET types (aka ”HINDER BY”, ecc..) allows the model to be able to grasp a more general sense of the dialogue. Given this premise, we have therefore also explored the first possibility by selecting the COMET proposals with minimum similarity scores (see Fig. 1, rightmost graph).

Although it is not as significant as in the random case, it seems that this selection criteria injects as much information, if not slightly more, than the authors' criteria or our previously explored criteria.

Since, however, also in this case the performance is not much better, and still far inferior w.r.t. the random case, we cannot find conclusive results about this new hypothesis, but deeper studies could lead to interesting conclusions. We can, however, conclude that which the COMET proposals are chosen does matter, but that choosing based on the above similarities might not be the best criteria.

### B. *Execution Time*

Even if we have reduced the datasets and the training epochs significantly, each run took about 2 hours, while the training of the shallow neural network took approximately 8 hours. In order to maximize our computational capabilities we have exploited Google Colab's T4 GPU. With each day-session we were able to run 1 full experiment run or partially compute the training of the similarities predictor starting from a checkpoint.

### C. *Main challenges*

This project presented multiple hurdles along its completion. In particular the low computational resources represented a huge challenge. Due to this constraint, we could only draw hints and not definitive results about our proposed method.

Nonetheless, the experimental results obtained in this restricted setting still might be useful in opening a new point of view in dialogue summarization task field.

## REFERENCES

1. Kim, S. *et al. Mind the Gap! Injecting Commonsense Knowledge for Abstractive Dialogue Summarization* 2022. arXiv: 2209.00930 `[cs.CL]`.
2. Bosselut, A. *et al. COMET: Commonsense Transformers for Automatic Knowledge Graph Construction* 2019. arXiv: 1906.05317 `[cs.CL]`.
3. Gliwa, B., Mochol, I., Biesek, M. & Wawer, A. *SAMSum Corpus: A Human-annotated Dialogue Dataset for Abstractive Summarization* in *Proceedings of the 2nd Workshop on New Frontiers in Summarization* (Association for Computational Linguistics, 2019). http://dx.doi.org/10.18653/v1/D19-5409.

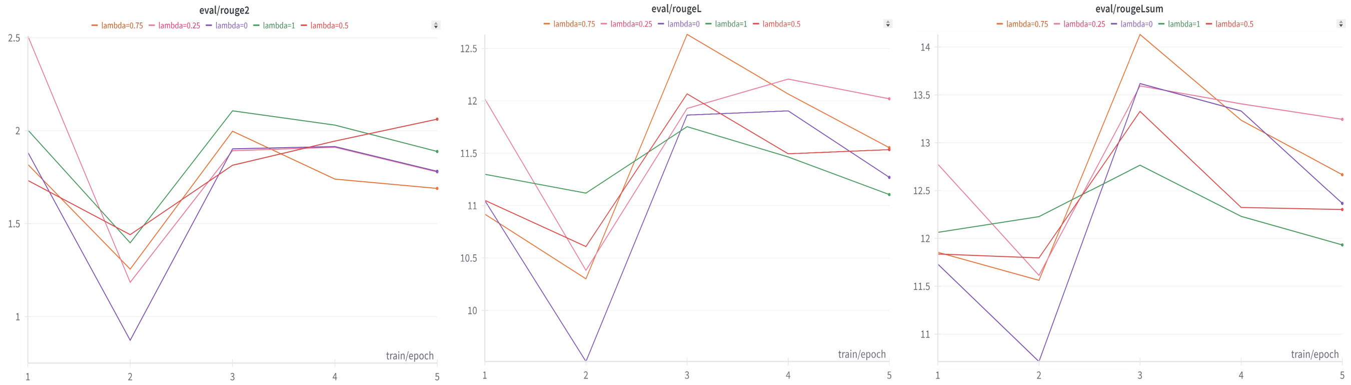
4. Hwang, J. D. *et al. COMET-ATOMIC 2020: On Symbolic and Neural Commonsense Knowledge Graphs* 2021. arXiv: 2010.05953 [cs.CL].
5. Lewis, M. *et al. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension* 2019. arXiv: 1910.13461 [cs.CL].
6. Zhou, P. *et al. RICA: Evaluating Robust Inference Capabilities Based on Commonsense Axioms* 2021. arXiv: 2005.00782 [cs.CL].
7. Chen, Y., Liu, Y., Chen, L. & Zhang, Y. *DialogSum: A Real-Life Scenario Dialogue Summarization Dataset* 2021. arXiv: 2105.06762 [cs.CL].
8. Li, Y. *et al. DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset* 2017. arXiv: 1710.03957 [cs.CL].
9. Sun, K. *et al. DREAM: A Challenge Dataset and Models for Dialogue-Based Reading Comprehension* 2019. arXiv: 1902.00164 [cs.CL].
10. Cui, L., Wu, Y., Liu, S., Zhang, Y. & Zhou, M. *MuTual: A Dataset for Multi-Turn Dialogue Reasoning* 2020. arXiv: 2004.04494 [cs.CL].
11. Reimers, N. & Gurevych, I. *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks* 2019. arXiv: 1908.10084 [cs.CL].

## IV. Appendix

In this appendix more detailed results about the ROUGE scores of our experiments are shown, in order to better understand the trend in the data.

### A. $\lambda$ experiments

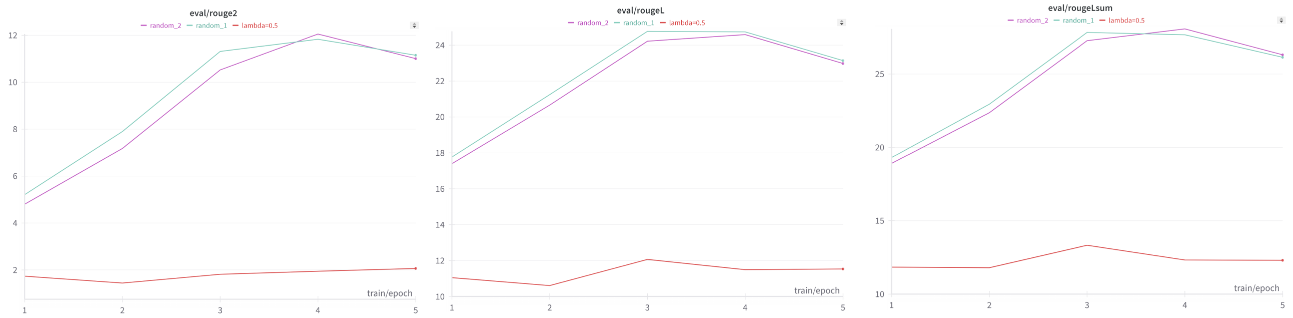
Below are shown the extensive results for the various configurations of the parameter $\lambda$:



Results for the experiments on the various configurations for the parameter $\lambda$

### B. Random experiments

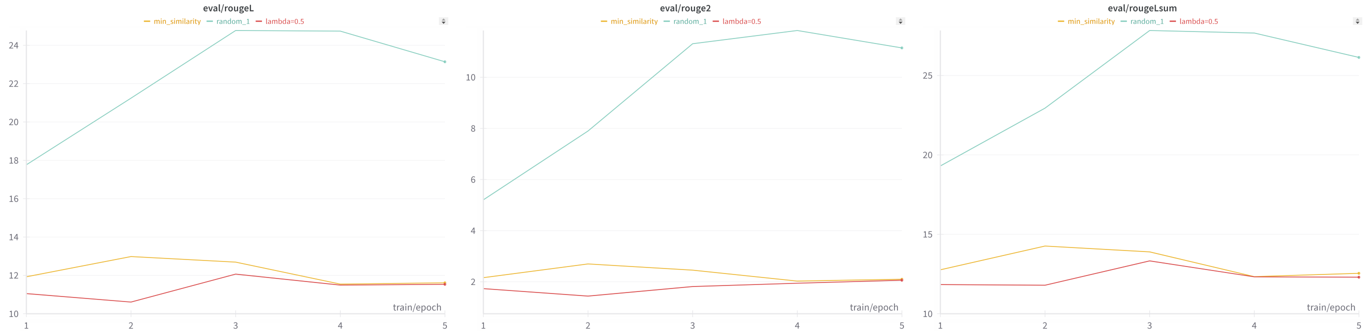
Below are shown the extensive results comparison of the experiments on $\lambda$ and the random selection of COMET proposals:



Results for the experiments on the random selection of COMET proposals, compared with the ones obtained with $\lambda$=0.5

### C. *Minimum similarity experiments*

Below are shown the extensive results comparison of the experiments on $\lambda$, the random and the minimum similarity selection of COMET proposals:



Results for the experiments on the random selection of COMET proposals, compared with the ones selected with the minimum similarity criteria