Less is More? Investigating Meta-Learning's Suitability in Sentence Compression for Low-Resource Data

L. Gustavo Coutinho do R.[1], **José Antônio F. de Macêdo**[1], **Ticiana L. Coelho da Silva**[1]

[1]Insight Data Science Lab – Universidade Federal do Ceará (UFC)

{gustavo.coutinho, jose.macedo, ticianalc}@insightlab.ufc.br

Abstract. *The sentence compression task is essential in the text summarization process. Unfortunately, the lack of labeled data for specific domains restricts the training of deep learning models to address this problem effectively. In this paper, we present an approach using a meta-learning algorithm called MAML to tackle this issue and assess the viability of this technique for the given task, with particular emphasis on its comparison to a fine-tuned BERT model. Our experiments reveal that a simpler approach involving fine-tuning a language model, such as BERT, might indeed be more effective in low-resource scenarios, consistently outperforming the meta-learning techniques for this particular task.*

1. Introduction

The dramatic increase in textual data on the Internet has made it challenging for users to extract valuable information in a reasonable time. Natural Language Processing (NLP) systems can help reduce this workload by performing tasks such as text classification, named entity recognition, and text summarization. In the context of text summarization, sentence compression plays a crucial role in generating concise yet meaningful summaries.

Sentence compression aims to create a shorter version of an input sentence while retaining essential information and ensuring grammatical correctness. There are two types of compressions: extractive and abstractive [Tas and Kiyani 2007]. Extractive compressions remove words without altering the word order, while abstractive compressions generate summaries by rearranging or introducing new words without restrictions.

Existing sentence compression models often rely on large-scale data, but specific domains and Low Resource Languages (LRL) face data scarcity challenges. To address this, we propose employing the few-shot learning paradigm to fine-tune pre-trained models using limited examples from different domains to simulate the low resource scenario. We utilize optimization-based meta-learning techniques, specifically the model-agnostic meta-learning (MAML) algorithm [Finn et al. 2017], to leverage existing models and datasets.

The main challenges in applying meta-learning techniques in sentence compression tasks are:

1. The existence of only one compression for each sentence and one sentence for each compression in the dataset used, unlike image classification tasks that have large datasets, such as Imagenet, and which have several examples for each class, such as cats, dogs, lions, and birds classes.

2. The existence of few datasets for extractive sentence compression, such as the Google News Dataset (GND) that we will use throughout this work, mainly for LRL such as Brazilian Portuguese.
3. Even the most popular dataset for the sentence compression task (GND) does not have any information about grouping the pairs of sentences into categories. Most papers that use meta-learning for few-shot learning tasks group the data for task creation [Mi et al. 2019, Yu et al. 2018].

To the best of authors' knowledge, no paper addresses these problems for the sentence compression task.

The main contributions of this paper include the following:

- We present a novel approach to addressing the sentence compression task by framing it as a Named Entity Recognition (NER) problem.
- We propose a method for modeling the sentence compression task as a meta-learning problem, utilizing few-shot learning principles and the well-established model-agnostic meta-learning (MAML) algorithm.
- We conduct comprehensive evaluations using the Google News dataset, exploring various few-shot scenarios, dataset divisions, and comparing with a BERT-based solution.

Our findings indicate that while models developed using the meta-learning approach can rapidly adapt to new tasks, they do not surpass the performance of a BERT model fine-tuned with the same dataset. This observation somewhat contrasts with findings in the context of other NLP tasks [Mi et al. 2019, Yu et al. 2018].

2. Related Works

Traditional Methodologies for Sentence Compression: Early approaches to extractive sentence compression relied on parsing tree pruning methods [Filippova and Altun 2013]. However, these techniques were prone to errors in constructing the trees themselves. Contemporary strategies began to frame the problem as Seq2Seq tasks, incorporating various modifications in LSTM-based models [Filippova et al. 2015, Soares et al. 2020, Kamigaito and Okumura 2020]. To the best of the author's knowledge, none of these methods effectively address the challenge posed by the limited availability of training data.

Meta-Learning in NLP Tasks: Meta-learning techniques have gained traction in various Natural Language Processing tasks [Lee et al. 2022]. One reason for the growing popularity of meta-learning is its effectiveness in low-resource situations, where collecting and annotating datasets can be prohibitively expensive, such as in Natural Language Generation [Mi et al. 2019, Qian and Yu 2019] and Machine Translation [Gu et al. 2018] tasks. Another factor driving the adoption of these techniques is the domain shift between training data and real-world testing and application scenarios [Li et al. 2020, Song et al. 2019]. The sentence compression problem addressed in this study faces both of these challenges.

3. Meta-Learning for Low-Resource Sentence Compression

One of the goals of this study is to develop an efficient model for the sentence compression task in low-resource scenarios, such as those found in medical or law enforcement

domains. To achieve this, we propose framing the task as a meta-learning problem, enabling the model to leverage knowledge from related tasks. The modeling process consists of four main steps:

1. Reformulate the sentence compression problem as a NER problem, utilizing a pre-trained BERT model as the foundation,
2. Create a dataset comprised of many sentence compression tasks,
3. Train a meta-model using the created tasks and the base BERT model,
4. Perform adaptation of the meta-model for a new sentence compression task, potentially within a distinct domain.

The following subsections provide a detailed description of each of these steps in the proposed methodology.

3.1. Sentence Compression as a NER problem

The NER problem is a well-established task in NLP, which involves identifying and classifying named entities within a given text. The goal of redefining sentence compression as an NER problem is to take advantage of the vast knowledge and tools available for NER and use them for the sentence compression task. By doing so, we aim to exploit the pre-existing strengths of BERT models in capturing contextual information and handling diverse linguistic structures for improved compression performance [Ma et al. 2019].

The process of task adaptation consists of assigning an entity label to each word in the dataset's compressions. We use two entity labels: `keep` for words that should be retained in the compressed sentence and `compress` for words that should be omitted. This transformation allows us to treat sentence compression as a NER task, where the objective is to identify and classify words in a sentence as either `keep` or `compress`.

To train a model to solve this adapted task, we employ a pre-trained BERT model as the foundation, given its proven success in various NLP tasks, including NER. We add an additional classification layer to the model to classify each word of each sentence as either `keep` or `compress`. The model is then fine-tuned on the newly created NER-style sentence compression dataset, learning to recognize and classify essential words and phrases to be retained in the compressed sentence.

3.2. Dataset creation

When modeling a problem using the meta-learning approach, we assume access to a distribution of tasks $\mathcal{P}(\mathcal{T})$. The objective of assuming this distribution is to sample T_i tasks from the $\mathcal{P}(\mathcal{T})$ distribution – where T_i is a task composed of a training set ($D_{T_i}^{Train}$, also known as support set) and a test set ($D_{T_i}^{Test}$, also known as query set) – to train a meta-model that can generalize well to all tasks used in the training process. The trained meta-model can then be fine-tuned to a task T', also sampled from the $\mathcal{P}(\mathcal{T})$ distribution, that was not seen in the meta-model training [Bansal et al. 2021]. When dealing with supervised learning tasks using meta-learning, we create the $\mathcal{P}(\mathcal{T})$ distribution based on a fixed set of tasks, subsampled from all classes [Vinyals et al. 2016].

In our sentence compression tasks, a sentence is an input x and its compression is an output y. An example task includes a Support Set (or D_{Train}) of five sentences with their compressions, and a Query Set (or D_{Test}) of five distinct sentences and their corresponding compressions from the Support Set. Mapped to the N-way, K-shot learning scenario, it yields values of $N = 5$ and $K = 1$.

3.3. Meta-training a model

Following the creation of the dataset, we utilized the Model-Agnostic Meta-Learning (MAML) algorithm to train a meta-model. This model is designed to process a range of sentence compression tasks as input. The meta-model we chose is an instance of the BERT model, which consists of approximately 107 million parameters.

The meta-model will be initialized with random θ^{Meta} parameters (also called meta-parameters). In an iteration step of the MAML algorithm, each task T_i will use a copy of all the parameters θ_i (here called task parameters) of the meta-model and will optimize them using the dataset $D_{T_i}^{Train}$, generating updated parameters θ_i'. Once all tasks are optimized, the datasets $D_{T_i}^{Test}$ will be used to optimize the meta-parameters θ^{Meta} using the parameters θ_i' of the tasks that were calculated. This way, the meta-model will be generalizing all tasks used as input in its own meta-parameters.

3.4. Fine-tuning process

With the meta-model trained from N-*way*, K-*shot* learning tasks, we obtain the parameters θ^{Meta} that should generalize the tasks used in the previous step. Given a new sentence compression task with a small dataset for training, we can use the newly trained meta-model for the fine-tuning process. The θ^{Meta} parameters will be used to initialize a new model and update it with the data from the new task. Since the new task has a small amount of associated data, it will take advantage of the already trained parameters of the meta-model as a good starting point for optimization.

4. Experiments

In this section, we will first formally present the Research Questions (RQ) that this work aims to answer:

RQ1. How does a model trained using meta-learning and conventional machine learning compare?
RQ2. What is the change in the performance of a meta-model if we increase the number of examples per class of each class?

In summary, the RQs presented try to evaluate if using meta-learning techniques is viable for the sentence compression problem.

We will also present the baseline and experiment settings, the dataset used and the data augmentation process applied, and finally, the results of the experiments performed.

4.1. Baseline and Model Settings

Since MAML is model agnostic, for all experiments performed in this work, we used as a baseline for the conventional machine learning training a BERT model [Kenton and Toutanova 2019] with an additional classification layer to classify the words with the entities `keep` or `compress`. For the MAML implementation, we used the Pytorch framework.

For all experiments performed, we considered two different settings:

- Scratch-BERTSC: Fine tune the BERT with only the target N-*way*, K-*shot* low-resource task.

- Meta-BERTSC: Train a meta-model using MAML and different N-*way*, K-*shot* low-resource tasks and then fine-tune it to an unseen task.

In the Meta-BERTSC setting, the number of tasks used to train the meta-model, i.e., tasks batch size, is one of the parameters that will be evaluated in the experiments, varing among 16, 32, 64 and 128. We setted $\alpha = 0.001$ and $\beta = 0.0001$. A last parameter evaluated in this experiment is the size of each task: we considerer a 5-*way*, 1-*shot* scenario and a 5-*way*, 5-*shot* scenario.

All models, for both Scratch-BERTSC and META-BERTSC, were evaluated in the target task. For the evaluation metrics, we used the ROUGE score variances, i.e., ROUGE-1, ROUGE-2 and ROUGE-L. We will also present the accuracy for the trained models. To assess the similarity between the compression ratio of system outputs and that of gold compressed sentences, we employed the delta compression (ΔC) metric, which is the difference between the system compression ratio and gold compression ratio [Kamigaito et al. 2018].

In terms of training times, the Scratch-BERTSC model outperformed the META-BERTSC approach. For a batch size of 16 tasks in a 5-*way*, 1-*shot* scenario, for example, Scratch-BERTSC completed training in 49 seconds, while META-BERTSC took 168 seconds. Similarly, for a larger batch size of 128 tasks, Scratch-BERTSC required 366 seconds, and META-BERTSC required considerably more time, 1326 seconds.

4.2. Google News Dataset for SC

The Google News Dataset [Filippova and Altun 2013], comprising 200,000 pairs of news article headlines and their compressed versions, is used for all experiments in this work. Despite the categorized nature of news articles, this dataset doesn't provide information about the pairs' categories.

We adapted the Google News dataset into a meta-learning framework using the pipeline from Subsection 3.2, dividing it randomly into 5-*way*, 1-*shot* learning tasks. This implies each task has examples from five different classes, with one example per class, considering each compression as a class with one sentence generating that specific compression.

Unfortunately, with the original Google News Dataset, it is not possible to vary the size of K in an N-*way*, K-*shot* learning sentence compression task since it is practically impossible for two or more non-related sentences to have the same compression, i.e., two or more examples with the same class. We used data augmentation techniques to work around this problem to create more sentences based on one single compression and evaluate the scenario of increasing the number of N and K in the meta-model training.

4.3. Compression Augmentation

For the compression augmentation process, we used a pre-trained BERT model [Kenton and Toutanova 2019] on English language with a masked language modeling (MLM) objetive[1].

For each compression of our dataset, we randomly place mask tokens between the words. We empirically choose to insert 10% of the number of words as masks in the

[1]The pre-trained model used is available at `https://huggingface.co/bert-base-cased`

compression (or one mask if the compression had less than ten words). With too many masks, the sentence generated might not be semantically correct. Once all masks are placed, we use the previously mentioned pre-trained model to replace them with actual correct words.

This whole process can be repeated to generate any number of sentences based on a single compression. Figure 1 presents an example of the compression augmentation process.

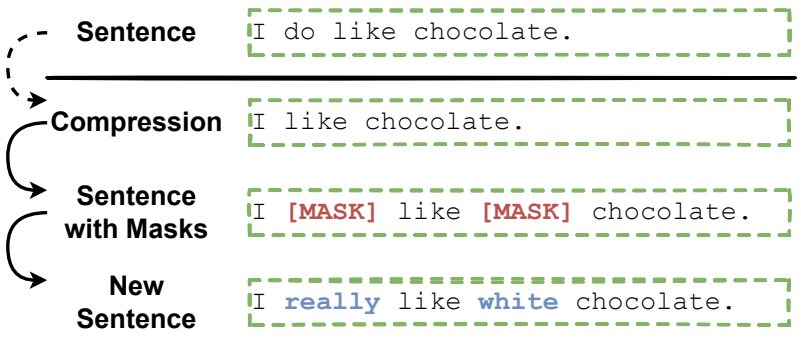

Figure 1. Example of a compression augmentation using a masked language modeling (MLM) objective.

After generating new sentences based on each compression, we are now able to create N-*way*, K-*shot* learning tasks with larger values of K: the task will have more than one sentence leading to the same compression (to do the reverse process of data augmentation, i.e., compress the sentence, we only need to extract the words added by the model to get the respective compression).

To verify that the generated sentences have the same meaning as the compressions that originated them, the sentence-level embeddings model Universal Sentence Encoder [Cer et al. 2018] were calculated for all sentences and compressions, and the values of each pair were calculated using the cosine similarity. We calculated the embedding similarities and extracted some descriptive statistics for the set of all comparisons: `mean=0.95440`, `median=0.96100`, and `standard deviation=0.03443`. These values show that the meaning of the newly generated sentence is very similar to the meaning of the original compression.

4.4. Results and Discussion

Meta-learning Versus Conventional Learning: To answer the RQ1., we compared two models created using the META-BERTSC and the Scratch-BERTSC settings described previously.

We trained the model from the META-BERTSC setting with 5-*way*, 1-*shot* randomly created tasks, i.e., we randomly chose five pairs of sentences and compression from the Google News Dataset to create each task. We vary the batch size of tasks from the list of values `[16, 32, 64, 128]`.

The rest of the dataset was used to sample sentence compression tasks to fine-tune the meta-model chosen previously individually. Table 1 presents the average F_1 Score

(F_1), ROUGE-1 (R-1), ROUGE-2 (R-2), ROUGE-L (R-L) and ΔC metrics of all models fine-tuned for `[16, 32, 64, 128]` tasks.

batch	F_1	R-1	R-2	R-L	ΔC
16	0.78002	0.76478	0.52067	0.76478	0.10163
32	0.77046	0.74560	0.50840	0.74522	0.08339
64	0.74938	0.72042	0.47680	0.72006	0.07716
128	0.78230	0.75714	0.52604	0.75606	0.08180

Table 1. Results for the 5-*way*, 1-*shot* learning scenario with the model trained from the META-BERTSC setting.

Table 2 presents the results of the model trained in the Scratch-BERTSC scenario with the same tasks and metrics used in the fine-tuning process for the META-BERTSC scenario described previously.

batch	F_1	R-1	R-2	R-L	ΔC
16	0.80777	0.77974	0.60439	0.77860	0.047612
32	0.80951	0.76883	0.59564	0.76835	0.046216
64	0.79398	0.75405	0.57421	0.75345	0.050612
128	0.80915	0.76810	0.59827	0.76682	0.041529

Table 2. Results for the 5-*way*, 1-*shot* learning scenario with the model trained from the Scratch-BERTSC setting.

The fine-tuned model Scratch-BERTSC consistently outperforms the META-BERTSC in all different metrics considered. Table 3 shows two examples of predicted compressions with the Scratch-BERTSC approach and the META-BERTSC approach.

Sentence:	Asda has dropped Saatchi & Saatchi out of the pitch for its £100m advertising account, ending its 20-year relationship with owner Publicis Groupe.
Compression:	Asda has dropped Saatchi & Saatchi for its £ 100m account.
[S-BSC]:	Asda has dropped Saatchi & Saatchi for its £100m account.
[M-BSC]:	Asda has dropped Saatchi Saatchi accounts for its £100m account.

Table 3. An example of compressions performed by the two models evaluated. In this table, the terms "S-BSC" and "M-BSC" are abbreviations for the terms "Scratch-BERTSC Prediction" and "META-BERTSC Prediction", respectively.

Increasing the Number K of Examples per Class: To answer RQ2., we compared two models created using the META-BERTSC setting: one with 5-*way*, 1-*shot* randomly created tasks (results from Table 1) and another with 5-*way*, 5-*shot* randomly created tasks. We also fine tuned the Scratch-BERTSC model with the same increased amount of tasks for the 5-*way*, 5-*shot* setting.

Table 4 and Table 5 presents the average F_1 Score (F_1), ROUGE-1 (R-1), ROUGE-2 (R-1), ROUGE-L (R-L) and ΔC metrics of all models fine-tuned for `[16, 32, 64, 128]` tasks.

batch	F_1	R-1	R-2	R-L	ΔC
16	0.75577	0.72926	0.49441	0.72828	0.06936
32	0.75175	0.71070	0.46656	0.70835	0.03720
64	0.75616	0.72122	0.48416	0.71966	0.05984
128	0.76329	0.73454	0.50208	0.73355	0.06201

Table 4. Results for the 5-*way*, 5-*shot* learning scenario with the model trained from the META-BERTSC setting.

batch	F_1	R-1	R-2	R-L	ΔC
16	0.80318	0.75739	0.59899	0.75519	0.02642
32	0.80239	0.76228	0.60694	0.75975	0.02743
64	0.79808	0.74732	0.57472	0.74573	0.03689
128	0.76638	0.71441	0.49862	0.71321	0.02050

Table 5. Results for the 5-*way*, 5-*shot* learning scenario with the model trained from the Scratch-BERTSC setting.

Based on the presented results, we cannot guarantee that a higher value of K generate better results, since for none of the two approaches are better than the other for all metrics, both for the Scratch-BERTSC and META-BERTSC approaches. Unfortunately there is a current limitation in the increase of the value of K because it would demand the generation of more sentences based on the same compression. The automatic addition of too many new words in the compression could generate syntactically incorrect sentences.

5. Conclusions and Future Works

In this work, we used the MAML algorithm for sentence compression via meta-learning, particularly when data is scarce. The resulting meta-model could be fine-tuned for specific domains with little data. However, our findings revealed its limitations, as it didn't outperform a fine-tuned BERT model, even with limited data. This suggests BERT's robustness and contextual understanding, combined with fine-tuning, may be a better approach for sentence compression in low-resource scenarios.

In future works, we intend to investigate the following additional research questions regarding the similarity between the distribution of tasks and the meaning bias between a sentence and the compressions generated during the work:

- RQ1 What would be the impact of training a meta-model with domain-specific tasks, e.g., sentence compression task with sentences only related to sports, and fine-tune it to a different domain task?
- RQ2 How similar are the tasks of the distribution used? And when we separate by domain, how similar are the tasks? And the new tasks that the meta-model will be tuned for, how similar are they to each other?

Finally, the present work can also serve as a basis for researchers who wish to investigate the advantages of leveraging pre-trained BERT models and fine-tuning techniques and the use of meta-learning in the sentence compression task since we believe this is the first study to propose this approach to this problem.

References

Bansal, T., Gunasekaran, K. P., Wang, T., Munkhdalai, T., and McCallum, A. (2021). Diverse distributions of self-supervised tasks for meta-learning in NLP. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5812–5824, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., et al. (2018). Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Filippova, K., Alfonseca, E., Colmenares, C. A., Kaiser, L., and Vinyals, O. (2015). Sentence compression by deletion with LSTMs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 360–368, Lisbon, Portugal. Association for Computational Linguistics.

Filippova, K. and Altun, Y. (2013). Overcoming the lack of parallel data in sentence compression. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1481–1491, Seattle, Washington, USA. Association for Computational Linguistics.

Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pages 1126–1135. PMLR.

Gu, J., Wang, Y., Chen, Y., Li, V. O. K., and Cho, K. (2018). Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3622–3631, Brussels, Belgium. Association for Computational Linguistics.

Kamigaito, H., Hayashi, K., Hirao, T., and Nagata, M. (2018). Higher-order syntactic attention network for longer sentence compression. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1716–1726, New Orleans, Louisiana. Association for Computational Linguistics.

Kamigaito, H. and Okumura, M. (2020). Syntactically look-ahead attention network for sentence compression. In *Proceedings of the AAAI*, volume 34, pages 8050–8057.

Kenton, J. D. M.-W. C. and Toutanova, L. K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Lee, H.-y., Li, S.-W., and Vu, T. (2022). Meta learning for natural language processing: A survey. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 666–684, Seattle, United States. Association for Computational Linguistics.

Li, J., Shang, S., and Shao, L. (2020). Metaner: Named entity recognition with meta-learning. In *Proceedings of The Web Conference 2020*, pages 429–440.

Ma, X., Xu, P., Wang, Z., Nallapati, R., and Xiang, B. (2019). Domain adaptation with bert-based domain classification and data selection. In *Proceedings of the 2nd DeepLo*, pages 76–83.

Mi, F., Huang, M., Zhang, J., and Faltings, B. (2019). Meta-learning for low-resource natural language generation in task-oriented dialogue systems. In *Proceedings of the 28th IJCAI*, pages 3151–3157.

Qian, K. and Yu, Z. (2019). Domain adaptive dialog generation via meta learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2639–2649, Florence, Italy. Association for Computational Linguistics.

Soares, F. M., da Silva, T. L. C., and de Macêdo, J. F. (2020). Sentence compression on domains with restricted labeled data. In *Proceedings of the 12th ICAART*, pages 130–140.

Song, Y., Liu, Z., Bi, W., Yan, R., and Zhang, M. (2019). Learning to customize language model for generation-based dialog systems. *CoRR*, abs/1910.14326.

Tas, O. and Kiyani, F. (2007). A survey automatic text summarization. *PressAcademia Procedia*, 5(1):205–213.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. (2016). Matching networks for one shot learning. *NeurIPS*, 29.

Yu, M., Guo, X., Yi, J., Chang, S., Potdar, S., Cheng, Y., Tesauro, G., Wang, H., and Zhou, B. (2018). Diverse few-shot text classification with multiple metrics. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1206–1215, New Orleans, Louisiana. Association for Computational Linguistics.