Mingcheng Wang, Eric Xia, Trevor Jung
University of Washington
{wmingch, ericxia, tjung2}@uw.edu

Reproducibility Summary

Scope of Reproducibility

We attempt to reproduce the claim that the authors' decision-focused summarization model, DecSum, substantially outperforms text-only summarization methods and model-based explanation methods in terms of decision faithfulness and representativeness, and that DecSum is the only method that enables humans to outperform random chance in predictions where restaurants will be better rated in the future.

Methodology

We used the author's code and ran DecSum on single Tesla T4 GPU Google Cloud Computing VM setups. Time of training and evaluation of results will be added later.

Results

TODO

What was Easy

TODO

What was Difficult

TODO

Communication with Original Authors

TODO

1 Introduction

We attempt to reproduce the experiments described in the EMNLP 2021 paper Decision-Focused Summarization by Chao-Chun Hsu and Chenhao Tan. [1] This paper proposes a new technique called decision-focused summarization for the problem of text summarization. Traditionally, text summarization is done only using textual information. However, this has issues. For instance, a common metric used to evaluate text summaries is non-redundancy, which encourages a summary to encompass a diverse range of topics. But if a doctor wanted to analyze the risks of pancreatic cancer in a patient, it may not be useful to include information about a knee injury. Instead, with a decision-focused summarization method, one could create summaries better tailored to a task.

The authors use the following setup: they leverage a supervised decision model, which takes as text as input and makes a decision. Then, the text summarization algorithm seeks to minimize loss on the decision. The authors propose two new desiderata: decision faithfulness and decision representativeness. In addition, they also use textual non-redundancy as a third desiderata.

The authors find that decision-focused summarization outperforms text-only summarization methods and model-based explanation methods. In addition, they show that the decision-focused summarization method outperforms other methods in aiding human decision-making in a challenging classification task.

2 Scope of Reproducibility

Parts we will be able to reproduce:
1. Run DecSum and apply it to Yelp data set to find its performance, including the decision faithfulness and representativeness.
2. Compare the results of the above with the statistics provided in the paper.
3. Compare DecSum's performance to text-only summarization and model-based explanation to see if DecSum indeed has better performance in terms of decision faithfulness and representativeness.

Parts we will not be able to reproduce:
1. The paper runs text-only summarization baselines (e.g., PreSumm, BART, Random) and model-based explanations (e.g. Integrated Gradients and Attention) to compare them with DecSum. Given our time and computing resources, we may rely on the results of text-only summarization baselines and Model-based explanations shown in the paper, instead of running those simulations by ourselves.
2. The paper implemented the human evaluation by using Amazon Mechanical Turk to solicit human guesses on which restaurants would be rated higher after participants. We are not able to implement such a high-scale data collection method like Amazon Mechanical Turk.

2.1 Addressed Claims from the Original Paper

Claims we are testing:

1. DecSum substantially outperforms text-only summarization methods and model-based explanation methods in terms of decision faithfulness and representativeness.

2. DecSum is the only method that enables humans to outperform random chance in predictions where restaurants will be better rated in the future.

3 Methodology

We used the author's code available on their Github, and documentation was available on their README.md on their Github. For computing resources, we used Google Cloud Computing with a single Tesla T4 GPU to run the DecSum experiments with 7 different hyperparameter permutations, as described both in Table 1 of the original paper and section 3.3 of this report.

3.1 Model Descriptions

The goal is to determine the most relevant information from a text for a particular decision to ultimately be a summary of sorts of human decision-making for a use-case. More formally, given an input text $X = \{x_s\}_{s=1}^{S=1}$, where S is the number of sentences, we wish to select a subset of sentences $\tilde{X} \subset X$ to support making a decision y. Thus, the training set, $D_{train} = \{(X_i, y_i)\}$, will be processed in such a way as to find defining traits in a text that lead to a certain decision y. The yelp datset is then defined as follows: for each restaurant, define X to be the text body of the first k reviews and let y be the average ranking of the first t reviews, where $t > k$ s.t. the problem is to predict future ratings. Here, the parameters are k and t; the researchers used $k = 10, t = 50$ s.t. their objective was to choose sentences amongst a restaurant's first 10 reviews that supported predicting its future review rating after 50 reviews.

3.1.1 The Loss Function

Loss functions were formalized from the need for the model to satisfy "decision faithfulness, decision representativeness, and textual non-redundancy", where $f : X \to y$ is the model's function.

Decision faithfulness. The sentences selected by the model should lead to the same decisions as the full input text: for selected sentences $\tilde{X} \subset X$, we want $f(\tilde{X}) \simeq f(X)$. A possible loss function is therefore the absolute difference between $f(\sim X)$ and $f(X)$, and the log of this is used for scalability:

$$\mathcal{L}_F\left(\tilde{X}, X, f\right) = \log |f(\tilde{X}) - f(X)|.$$

Decision representativeness. We want the summary of the decision distribution outputted by teh model, $\hat{Y}_{\tilde{X}} = \{f(x) \mid x \in \tilde{X}\}$, to be close to the decision distribution of all sentences from the full text, $\hat{Y}_X = \{f(x) \mid x \in X\}$. The researchers used the Wasserstein Distance to measure the distance between $\hat{Y}_{\tilde{X}}$ and $\hat{Y}_X$:

$$W(\hat{Y}_{\tilde{X}}, \hat{Y}_X) = \inf_{\gamma \in \Gamma(\hat{Y}_{\tilde{X}}, \hat{Y}_X)} \int_{\mathbb{R}\times\mathbb{R}} ||f - f'|| d\gamma(f, f'),$$

where $\Gamma(\hat{Y}_{\tilde{X}}, \hat{Y}_X)$ is the collection of all measures on $\mathbb{R} \times \mathbb{R}$ with marginals $\hat{Y}_{\tilde{X}}, \hat{Y}_X$ on the first and second factors, respectively. Again, the log of this was used for scalability, such that the second loss function for the model was:

$$\mathcal{L}_R(\tilde{X}, X, f) = \log\left(W(\hat{Y}_{\tilde{X}}, \hat{Y}_X)\right).$$

Textual non-redundancy. As the model's objective is a summary of the reasoning for the writer's decision, we use cosine similarity to encourage dissimilar sentences in the output summary:

$$\mathcal{L}_D(\tilde{X}) = \sum_{x \in \tilde{X}} \max_{x' \in \tilde{X} - \{x\}} \text{cossim}(s(x), s(x')).$$

Final loss function. Combining these three, out loss function is

$$\mathcal{L}(\tilde{X}, X, f) = \alpha\mathcal{L}_F(\tilde{X}, X, f) + \beta\mathcal{L}_R(\tilde{X}, X, f) + \gamma\mathcal{L}_D(\tilde{X}),$$

where α, β, γ are parameters that determine the weight of each component of the loss function.

3.1.2 The Model

We use an iterative algorithm that greedily selects a sentence that minimizes the loss function: In order to select K sentences from an input X, in each step $k = \{1, \ldots, K\}$, iteratively choose a sentence from the remaining sentences $\hat{x} \in X - \tilde{X}_{k-1}$ that achieves the lowest loss $\mathcal{L}(\tilde{X}_{k-1} \cup \{\hat{x}\}, X, f)$ where $\tilde{X}_{k-1}$ is the current summary using $k - 1$ sentences. When β is positive, only $\mathcal{L}_R$ is used in the first step "to encourage the algorithm to explore the full distribution rather than stalling at the sentence most faithful to $f(X)$". In practice, beam search with beam search size 4 was used to improve the greedy algorithm.

A 64%/16%/20% training/validation/test set split was employed, and Longformer [2] (Github code here) was used to fine-tune the regression models.

Algorithm 1: `DecSum`

Input: X, f, K
Output: $\tilde{X}$
$\tilde{X} \leftarrow \emptyset,\ k \leftarrow 1;$
while $k \leq K$ **do**
  if $\beta > 0$ *and* $k = 1$ **then**
    $\hat{x} \leftarrow \text{argmin}_{\hat{x} \in X} \mathcal{L}_{\text{R}}(\{\hat{x}\}, X, f)$
  else
    $\hat{x} \leftarrow \text{argmin}_{\hat{x} \in X} \mathcal{L}(\tilde{X} \cup \{\hat{x}\}, X, f)$
  end
  $\tilde{X} \leftarrow \tilde{X} \cup \{\hat{x}\};$
  $X \leftarrow X - \{\hat{x}\};$
  $k \leftarrow k + 1$
end

3.2 Datasets

We used the Yelp dataset, provided freely on the Yelp website.

3.3 Hyperparameters

One hyperparameter group we used was what values to use for the three desiderata (decision faithfulness, decision representativeness, textual non-redundancy) when implementing DecSum. We ran our experiments using the same as the ones given in Table 1 of their paper, as reformatted below. Note that "**Full**" in the table refers to using all the reviews without summarization, and **MSE** is the mean-squared error.

`DecSum` with (α decision faithfulness, β decision representativeness, γ textual non-redundancy)

Method	MSE with Full (faithfulness)	MSE
(1, 1, 1)	0.0005	0.136
(1, 1, 0)	0.0378	0.164
(1, 0, 1)	0.0002	0.135
(0, 1, 1)	0.162	0.283
(1, 0, 0)	0.0264	0.155
(0, 1, 0)	0.175	0.287
(0, 0, 1)	0.504	0.565

3.4 Implementation

We use the author's code, provided on their Github. Python and bash shell script were used; the python packages used were {argparse, collections, datetime, glob, gzip, json, logging, nltk, numpy, os, pandas, pathlib, pickle, pprint, pytorch_lightning, scipy, sentence_transformers, sklearn, spacy, summa, time, torch, tqdm, typing, wandb}.

3.5 Experimental Setup

TODO

3.6 Computational Requirements

The authors preprocess their data and train their model on a single RTX 3090 setup. The performance of a Tesla T4 seems to lag behind an RTX 3090, tho not overly so as one would expect from, say, a 3090 and a 1080ti, and so we expect that the GPU hours we will spend on training and processing the data to be a few hours more than that of the authors. For reference, the authors spent an hour per epoch on fine-tuning the regression models before DecSum was ran, and less than an hour on DecSum itself. The memory requirements shouldn't be insane, as the yelp dataset itself is only 13 GB.

4 Results

TODO

4.1 Result 1

4.2 Result 2

4.3 Additional Results not Present in the Original Paper

TODO

5 Discussion

TODO

5.1 What was Easy

TODO

5.2 What was Difficult

TODO

5.3 Recommendations for Reproducibility

TODO

Communication with Original Authors

TODO

References

[1] Chao-Chun Hsu and Chenhao Tan. Decision-focused summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 117–132, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[2] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2020.