Reproducibility Report - Decision-Focused Summarization

Madison Doerr **Shweta Mayekar** **Kirsteen Ng** **Roger Wang**
University of Washington
{mcdoerr, shwet695, kirstng, rogerwyf}@uw.edu

Reproducibility Summary

Scope of Reproducibility

The primary goal of the paper "Decision-Focused Summarization" [1] is to summarize relevant information from textual data in order to optimize the decision making process through text summarization. This is achieved by DecSum, the proprietary approach developed by the authors. The paper claims to utilize restaurant reviews from Yelp and summarize the reviews with DecSum to predict future ratings for those restaurants by leveraging a regression model. It demonstrates how a decision can be inferred from text. The paper claims that DecSum substantially outperforms other summarization baseline methods and enables humans to make better decisions.

Methodology

We used the existing code by the authors and developed additional scripts in Python to reproduce all the experiments in this report. Using Yelp's data on restaurant reviews we first demonstrated how Longformer as a regression model can properly distinguish sentences located at different ranges of sentiments. To demonstrate how a decision can be inferred from text we ran DecSum which summarized the first 10 reviews. We then predicted a restaurant's future rating by leveraging Longformer with the summarized first 10 reviews as model input and compared DecSum to Random Selection as the baseline summarization method. Our evaluation consisted of both formulaic metrics and human evaluations. We used Mean Squared Error (MSE) on the restaurant rating after 50 reviews to measure the quality of the summaries in the forecasting task. Re-implementation of this pipeline helped investigate human-centered approaches to support human decision making. We ran these experiments on the CSE GPU machine, nlpg00 with a GTX 1080 Ti (11G), and it took 56 minutes to train Longformer and 145 hours to train DecSum and run it on the test data.

Results

We have successfully verified our claims/hypotheses and the following table summarizes our results.

Hypothesis/Claim	Result
For a regression task with text data as inputs, Longformer leads to an appropriate distribution of the predicted score $f(x)$.	The tri-modal pattern in the distribution plot of Longformer predictions shows that the model is able to distinguish reviews with positive, neutral and negative sentiments across all groups.
DecSum outperformed Random Selection in summarizing text data for a regression task.	With Longformer model as the regression model, DecSum outperformed Random Selection by 0.12 in MSE on test set of Yelp restaurant review data.
DecSum review summary allows human evaluation on future restaurant ratings.	DecSum summary assists human evaluation and outperforms random selected summary by 13% in accuracy in a task of predicting better restaurants.

What was Easy

The instruction on the GitHub repository to replicate the experiment was very clear and easy to follow. The authors' codes were well documented. The concept behind DecSum was easy to understand.

What was Difficult

The Longformer model was difficult to understand as we needed to modify the codes to test with our random sentences sample. Time estimation for running the model on our hardware was complicated to achieve.

Communication with Original Authors

We contacted the main author in order to obtain the original dataset and inquired details about the models mentioned in the original paper.

1 Introduction

The original research creates a decision-focused summarization method, named **DecSum**, which aims to extract representative sentences from the full text body to support humans make better decisions. In addition to the textual non-redundancy method, this summarization technique incorporates two important and novel desiderata, decision faithfulness and decision representativeness to extract relevant information from texts. This approach is claimed to substantially outperform text-only summarization and model-based explanation methods in decision faithfulness and representativeness.

Definitions of three major desiderata of DecSum in this paper:

- *Decision faithfulness*: A measurement that the selected sentences should lead to the same decision as using the full text based on the model.
- *Decision representativeness*: A measurement of how close the decision distribution of the summary is to the decision distribution of all sentences in the full text.
- *Textual non-redundancy*: A measurement of the diversity in textual summarization.

The experiment is performed via a future rating prediction task using Yelp text data. The task is to predict a restaurant's future rating by leveraging a regression model with the summary of its first 10 reviews by different summarization methods as the model input. DecSum is being evaluated against text-only summarization and model-based explanation baseline methods. For text-only methods, the baseline models are PreSumm(an extractive summarization method with hierarchical encoder), BART(a seq2seq model trained with a de-noising objective)and a Random method (randomly extracts sentences from the text body). For model based explanation techniques, the experiment has chosen Integrated Gradients and Attention as comparison attributes to be incorporated into the regression model.

2 Scope of Reproducibility

This research has made the following contributions:

- Proposed a novel summarization methodology that uses a greedy algorithm to iteratively search for the most representative sentences in a full text.
- Introduced decision faithfulness and representativeness as important desiderata for text summarization in addition to textual non-redundancy.
- Experimented on predicting future Yelp ratings using summarized data by leveraging a predictive model, which showed that DecSum outperformed several text-only and model-based baseline methods in both evaluation metrics and human accuracy.

We will replicate model experiments using Yelp dataset and compare DecSum to Random Selection as the baseline method. We will also replicate human evaluation experiments with the same setup described in the paper.

2.1 Addressed Claims from the Original Paper

The addressed claims from the original paper in our project are:

Claim 1. For a regression task with text data as inputs, Longformer leads to an appropriate distribution of the predicted score $f(x)$ at the sentence level, suggesting that Longformer model can properly distinguish sentences located at different score range.

Claim 2. DecSum outperformed Random Selection in summarizing text data for a regression task. The Mean Squared Error (MSE) with DecSum summary on test data as inputs for a Longformer model is lower than the one with Random Selection by 0.3.

Claim 3. DecSum enables humans to statistically outperform random chance in predicting which restaurant will be rated better in the future. This is measured by human accuracy in a binary classification task on predicting the restaurant with higher future rating based on a pair of summarized reviews of two restaurants.

3 Methodology

3.1 Model Descriptions

The main approach proposed by the paper, DecSum, is a method of summarization conditioned on a decision of interest. Given an input text $X = \{x_s\}_{s=1}^{s=S}$, where S is the number of sentences, the objective is to select a subset of sentences $\tilde{X} \subset X$ to support making the decision y. The DecSum approach contains two major components:

First, a supervised regression model is developed to make the decision y given input text data X. This paper claimed that models including Longformer, Logistic Regression and Deep Averaging Networks were considered for this regression task, and Longformer was selected as the final model, denoted as f below, for its better ability to generalize on shorter inputs and to distinguish sentences located at different score range.

Second, the core of DecSum is a greedy algorithm that selects K sentences from input X with f as its decision function. In each step $k \in \{1, .., K\}$, it iteratively chooses a sentence among the remaining sentences 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 with $k - 1$ sentences.

The aforementioned loss function $\mathcal{L}()$ for the algorithm is defined as a weighted sum of losses regarding three desiderata:

- *Decision faithfulness*: $\mathcal{L}_F(\tilde{X}, X, f) = \log|f(\tilde{X}) - f(X)|$. This measures how similar the decision made with the selected sentences is to the one made with full text.
- *Decision representativeness*: $\mathcal{L}_R(\tilde{X}, X, f) = \log(W(\hat{Y}_{\tilde{X}}, \hat{Y}_X))$, where $\hat{Y}_A = \{f(x) \mid x \in A\}$ and $W()$ is Wasserstein Distance. This measures how well the model decisions of selected sentences represent the decision distribution of sentences in the full input.
- *Textual non-redundancy*: $\mathcal{L}_D(\tilde{X}) = \sum_{x \in \tilde{X}} \max_{x' \in \tilde{X} - \{x\}} cossim(s(x), s(x'))$ where $s()$ is the SentBERT sentence representation. This encourages sentences in the summary to be dissimilar to each other.

Therefore the 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 hyperparameters $\alpha, \beta, \gamma \in \{0, 1\}$ control the tradeoff between the three considerata.

When $\beta > 0$, the algorithm only uses $\mathcal{L}_R$ at the first step to encourage the algorithm to explore the full distribution rather than stalling at the sentence that is most faithful to $f(X)$. In practice, the paper uses beam search with beam size of 4 to improve this greedy algorithm.

3.2 Datasets

The raw dataset can be downloaded from https://www.yelp.com/dataset/download according to the paper. This dataset contains reviews and their rating score (from 0 to 5) for all businesses listed on Yelp. **Note that this dataset has been updated by Yelp since the release of the paper**, thus while having the same format, the data are different from and the size is bigger than what the authors used in their original experiment.

This raw dataset is then preprocessed using `yelp_preprocess.py.` We filtered the dataset to select only the businesses that are identified as restaurants. There are **23,821** restaurants (compared to **18,112** in the original dataset) in total in the filtered dataset, and we used 64%/16%/20% training/validation/test split as described in the paper. These datasets are available under `data/out/50reviews` directory on our Github repository: https://github.com/mcdoerr/decsum.

3.3 Hyperparameters

3.3.1 Longformer

The Longformer model training details were pulled from the existing codebase. This model is trained with AdamW optimizer with learning rate of 5×10^{-5}, linear warm-up of 500 steps, batch size of 4 and maximum input token length of 3000. The hyperparameters of this model are number of epochs of {3, 4, 5} and max sequence lengths of {2000, 3000}, and the model with lowest MSE on the validation set is then chosen to be the final model.

However, in our case, the memory requirement for the GPU was too high to preserve the original hyperparameter for sequence length, so we reduced it to {50, 100} for training purposes so that the longformer model can fit into memory. This causes a performance drop on the validation set, which we will take into account in the following sections.

3.3.2 Ridge Regression

We noticed from the author's implementation that the **Logistic Regression model stated in the original paper is, in fact, a Ridge Regression model**, and thus it will be referred as Ridge Regression in the following sections. The best hyperparameter of the model, α the regularization strength, is found by searching through a geometric sequence from 0.03125 to 8 with ratio of 2 for the value with the lowest MSE on the validation set.

3.3.3 DecSum

The best hyperparameters for the DecSum algorithm, α, β, γ, are found using a grid search in $\{0, 1\}$ for each hyperparameter to find the combination with the lowest loss $\mathcal{L}$ on the validation set.

3.4 Implementation

For model training and evaluation, we used existing code and updated it to use a new dataset, which required changes to the preprocessing steps. The languages used are Python for the models and bash scripts for startup. Packages include PyTorch, Scikit-learn, and NumPy, etc.

For model comparison between Longformer and Ridge Regression, we developed a Python script to compute MSE of predictions from each trained model on test data and plot their distribution to verify Claim 1 that we listed above.

All codes used for experiments in this report can be found at our Github repository, forked from the original repository of the paper with additional code modification and development:
https://github.com/mcdoerr/decsum

3.5 Experimental Setup

3.5.1 Experiment 1 - Regression Model Training & Comparison

To verify Claim 1, we trained the two regression models (Longformer, Ridge Regression) involved in our addressed claim: For each restaurant in the training data, we used the full text of its first 10 reviews as input X and its average rating after 50 reviews as label y to fine-tune the regression models with AutoTokenizer and TfidfVectorizer to transform the input text data for Longformer and Ridge Regression, respectively.

With the two fitted models, we first generated predictions on the test data with full text as inputs, then followed the same method described in the paper to group these predictions into four groups where restaurant's average ratings for the 10 reviews are in [1.5, 2.5), [2.5, 3.5), [3.5, 4.5), and [4.5, 5] as group 2, 3, 4, and 5 respectively. The distribution of predictions from each model in each group was then plotted for the model comparison purpose.

3.5.2 Experiment 2 - Model-based Evaluation

To verify Claim 2, we generated predictions of average ratings after 50 reviews using the fitted Longformer model on the summarized test datasets by DecSum and Random Selection separately as we describe below:

- *DecSum* : For each restaurant, DecSum selects the best $K = 15$ sentences from the full text of its first 10 reviews as a summary of reviews with the trained Longformer model as its decision function f, then sequentially select sentences from the summary until the length exceeds 50 tokens to make the summary comparable, as stated in the paper.

- *Random Selection* : For each restaurant, Random Selection method simply randomly chooses sentences from the full text of its first 10 reviews. To make the summary comparable to the one from DecSum, we set the ratio of sentences selected from the full text of reviews to 30%.

With the predictions from each method, we then computed their corresponding Mean Square Errors against that actual average ratings after 50 reviews.

3.5.3 Experiment 3 - Human Evaluation

To verify Claim 3, we set up a simplified binary classification task for human evaluation following the same framework developed in the paper: A participant was first given a pair of reviews summarized by the same method from two different restaurants, then asked to guess which restaurant would be rated better after 50 reviews. We then checked their guesses with the actual average rating after 50 reviews of each restaurant to see if they were correct.

The sample data we gave out to the participants were composed with the same criteria as mentioned in the paper: Two restaurants in each pair are ensured to have the same average rating of their first 10 reviews, and their average ratings after 50 reviews are differed by at least 1. A total of 30 pairs of restaurants were picked in our sample data.

Due to time limitations, we were able to find 12 people to participate into this experiment. To reduce subjective bias from our participants, we asked each person to perform the binary classification task for only 5 pairs of restaurants, therefore 6 people were given pairs of reviews summarized by DecSum and the other 6 were given pairs of reviews summarized by Random Selection.

3.6 Computational Requirements

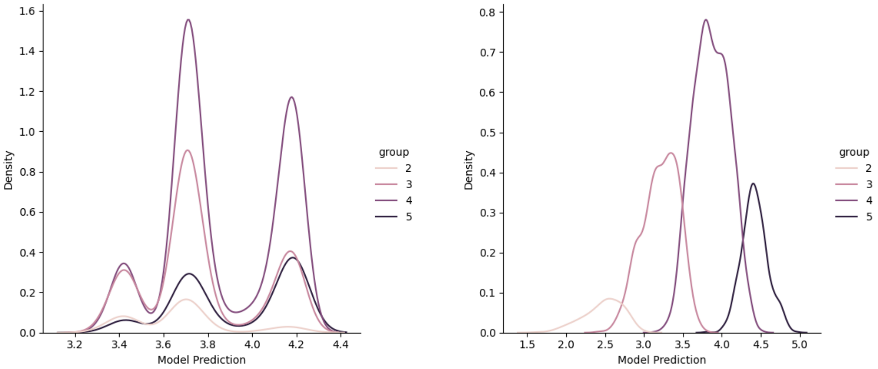
Our estimations for running our experiments initially were that it would take one GPU hour per epoch to train the Longformer model (thus 3 hours in total, since it was claimed in the paper that epoch = 3 is the best value for this hyperparameter) and 3 days to train and run DecSum algorithm on the test data with all three components ($\alpha = 1, \beta = 1, \gamma = 1$). These estimations were pulled from the paper in which RTX 3090 GPU was used for the original experiments.

Due to hardware limitations, We ran our experiments on the CSE GPU machine, nlpg00 with a GTX 1080 Ti (11G). As mentioned above in Section 3.3, we reduced max sequence lengths largely so that the longformer model can fit into memory. Because of this change, it took 56 minutes in GPU time to run the Longformer on the GTX 1080 Ti. The DecSum algorithm took 145 hours and 32 minutes in GPU time to train and run on the test data because of a bigger dataset we had than the original one authors used for their paper as well as our GPU memory bottleneck.

4 Results

4.1 Result 1 - Regression Model Comparison

With full text as input data, Longformer with epoch = 3 and max sequence lengths = 100 and Ridge Regression with $\alpha = 2$ are the best models from hyperparameter search based on the validation set. Since the subject of interest in this experiment is the distribution of predicted ratings, we show the distribution plots of model predictions in each group from the two models in Figure 1. These plots show similar patterns as in Figure 7 from the original paper.



Longformer

Ridge Regression

Figure 1: Model prediction distributions of each rating group from Longformer and Ridge Regression

It can be seen from the tri-modal pattern in the distribution plot of Longformer predictions that the model is able to distinguish reviews with positive, neutral and negative sentiments across all groups, whereas for Ridge regression, the predicted ratings are all approximately normally distributed with different mean values for all groups as described in the paper. This result verifies Claim 1 that Longformer leads to an appropriate distribution of the predicted score $f(x)$ compared to Ridge Regression since it can properly distinguish sentences located at different score range.

4.2 Result 2 - Model-based Evaluation: DecSum vs. Random Selection

With the fitted Longformer model from Result 1 as the decision function f, DecSum with $(\alpha, \beta, \gamma) = (1, 1, 1)$ is the best model from hyperparameter search. We then summarized reviews in test data with DecSum and Random Selection separately as described in 3.5.2 and 3.5.3, and evaluated the two summarization methods as described in 3.5.4.

Summarization Method	Test MSE w/ Longformer
DecSum	0.354
Random Selection	0.473

Table 1: MSE of Longformer predictions based on summaries of different methods.

In Table 1 we report the MSE of model predictions based on summaries of test data. The result for Random Selection aligns with the one from the original paper (0.473 vs. 0.475). However, our MSE from DecSum is higher than the result from the paper (0.354 vs. 0.136). We suspect that this discrepancy was due to the fact that our Longformer model was trained with a reduced maximum sequence length which might have resulted in a degraded performance of DecSum using it as the decision function.

Nevertheless, DecSum still outperformed Random Selection by 0.12 in MSE, indicating a better performance of DecSum in document summarization tasks. These results partially verify Claim 2.

4.3 Result 3 - Human Evaluation: DecSum vs. Random Selection

Following the experiment setup that we described in 3.5.5, in Table 2 we report the human accuracy of the experiments with 12 participants on 30 pairs of restaurants.

Summarization Method	# Correct Guess	Human Accuracy(%)
DecSum	27	90.0
Random Selection	22	73.3

Table 2: Performance of summarization methods on the simplified classification task with human evaluation

Our results for both methods are higher than those in the original paper (90% vs. 85.5% for DecSum, 73.3% vs. 58% for Random). With these numbers, we performed a one-sided, two-sample proportion test for the statistical significance of the better performance of DecSum than Random Selection in human accuracy. The chi-squared statistic from this test is $\chi = 2.787$ with the associated p-value $p = 0.048$, which indicates that the human accuracy with DecSum is statistically better than the one with Random Selection. This result verifies Claim 3 that DecSum enables humans to statistically outperform random chance in predicting which restaurant will be rated better in the future.

4.4 Additional Results not Present in the Original Paper

4.4.1 Additional Dataset

As mentioned in Section 3.2, our experiments were performed on the latest version of Yelp dataset which was different from and bigger than the one used for the paper since the authors did not publish the original datset on their github repository, thus extra amount of time and work were devoted into preprocessing our dataset for model training and other tasks in our experiments.

4.4.2 Lasso Regression

In many machine learning problems, regularization for linear models is usually applied as a technique to avoid overfitting. Compared to Ridge Regression used in this paper, Lasso Regression is known for its nice property to perform algorithmic feature selection to filter out unimportant features in the input vector. Therefore, we would like to see if a Lasso Regression is able to focus on only the words that contribute the most to the sentiment of a review in the input transformed by the Tfidf-Vectorizer instead of using all words with different weights in Ridge Regression.

We fine-tuned a Lasso Regression model with the same setup as described in 3.5.1. The best model based on validation set has a regularization strength $\alpha = 0.03$. In Figure 2, we plot the distribution of predictions from this fitted model on the test data.

As shown in the plot, not surprisingly, Lasso Regression was not able to distinguish sentences at different sentiments or score range. However, it does help with shrinking the range of predictions: In Ridge Regression, the model can make predictions beyond 5 stars whereas in Lasso Regression all predictions are within the range of (0, 5).

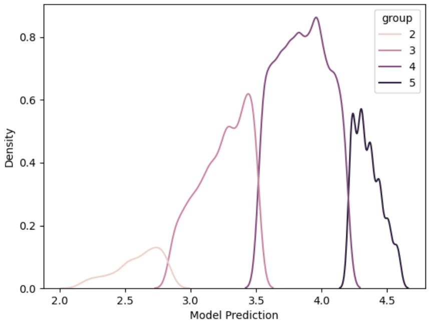


Figure 2: Model prediction distributions of each rating group from Lasso Regression

4.4.3 Additional Human Evaluation

As discussed in Section 3.5.5, one important rule in composing the sample data used for human evaluation experiments is that two restaurants in the same pair for comparison have average ratings after 50 reviews differed by at least 1. We suspect that this will artificially lead to a better performance of DecSum, which we will next explain in details.

After inspecting the actual reviews of restaurants, there seems to be a pattern for the restaurants that initially have a high average rating of the first 10 reviews and a low average rating after 50 reviews: The distribution of first 10 ratings tend to be left skewed, with one or two significantly low rating scores (e.g [4, 4, 5, 4, 5, 4, 1, 4, 1, 4]), whereas it is very rare for a restaurant with an initial low average rating to improve eventually. Therefore, DecSum seems to work very well at picking up sentences with negative sentiment among the majority of positive sentences. Below we show an example of DecSum summary of the first 10 reviews of a restaurant that has 4.0 average rating of the first 10 reviews and 2.8 average rating after 50 reviews.

> so I thought I would try it. Runny eggs, hockey puck sausage and all the dust, dirt and being yuckiest place was just too much. Definitely not going back for any meal. The butter almost didn't melt in the grits. Went there for dinner and we both had steak ..

Even though the average rating of the first 10 reviews (4.0) is relatively positive, DecSum picks mostly negative sentences as its summary, since the decision function f was trained by using the average rating after 50 reviews (2.8) as label.

Because of this observation, we would like to see DecSum's performance after relaxing the rule of comparing restaurants when their future ratings are differed by 1. We sampled additional 15 pairs of restaurants without this rule and asked 2 participants to evaluate all of them. We report the results in Table 3 below:

Participant	# Correct Guess	Human Accuracy(%)
Participant 1	12	80.0
Participant 2	13	86.6
Total	25	83.3

Table 3: Performance of DecSum with additional human evaluation

We performed the same hypothesis testing comparing the human accuracy from this experiment with the previous DecSum result and calculated $p = 0.074$. Although we cannot conclude DecSum performs statistically worse on such comparisons, there is some evidence indicating DecSum tends to help humans make better decisions when the two restaurants are more qualitatively different.

Noticeably, for one comparison where the restaurants have future rating differed by 0.2, both participants failed to guess the better restaurant. We attach this comparison in Table 4.

Summary - Restaurant 1 (avg. rating after 50 review = 4.94)	Summary - Restaurant 2 (avg. rating after 50 review = 4.74)
Typical Asian Tibetan food. Say hi to Kelly the friendly owner. Curry was a tad heavy on the salt, but the flavour is all there. We tried Chicken steam momo, Veg momo and Chicken Thupka. Both were delicious! We'll definitely come back and order that again!	Save room for their desserts! Yummy!! Can't wait to visit again! They custom-made a veggie risotto that was amazing!! Nice selection of main courses. A quick search found Sam's I had the chicken parmesan and my buddy had the shrimp scampi.

Table 4: Example - DecSum summary of first 10 reviews of two restaurants

5 Discussion

Overall we were able to replicate the experiments to verify several claims in the original paper. The fact that we have used a different Yelp dataset and was able to produce similar results showed that DecSum is able to generalize well for data within the same domain.

5.1 What was Easy

The instruction on the Github repository of the paper is very clear from the beginning: data source, preprocessing procedure, setting up the required environment and how to train the Longformer and DecSum. Although the codebase requires some technical changes to work, the authors did a good job at writing organized and excellent codes for this paper.

The concept behind DecSum was also easy to follow since the authors did a great job at explaining the three desiderata, the algorithm and its loss function for this approach.

5.2 What was Difficult

It took us some time at the beginning to understand what the longformer model was actually doing and how to modify the model so it can be fit into our hardware. Moreover, it was really difficult to estimate how long the model would take to run on our hardware and to write code to make sure that the training process would save checkpoints properly since the runtime was long as we described in Section 3.6.

5.3 Recommendations for Reproducibility

We recommend the authors including the original preprocessed datasets in the github repository so that others can replicate the experiments in the original paper with exactly the same data sources it used. In addition, technical details for all the baseline models and summarization methods mentioned in the paper were not shared in the repository (In fact, the authors left this section as "cleaning" in their readme file so we assume it is still work in progress), so we recommend the authors add these details if possible.

Communication with Original Authors

After realizing that our dataset from Yelp was different from the one used for this paper, we emailed the main author of the paper, Chao-Chun(Joe) Hsu at the University of Chicago, and decided to continue working on what was available to us while waiting for his response. Later on, Joe friendly replied to us with the original datasets as well as details of the logistic regression that we wanted to look into. However, since the model training process took very long as we mentioned in Section 3.6, we would not be able to finish the project in time if we were to switch our data source to the original dataset. Nevertheless, we would like to thank Joe for his response and friendly communication with us.

References

[1] C.-C. Hsu and C. Tan, “Decision-focused summarization,” in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 117–132. [Online]. Available: https://aclanthology.org/2021.emnlp-main.10