T^3: Train, Transform, Translate

Homer Walke, Gabriel Marks, Sebastien Jean-Pierre, David Cabatingan

CSCI 1470 Fall 2019: Group Big Brain Learning 🤔

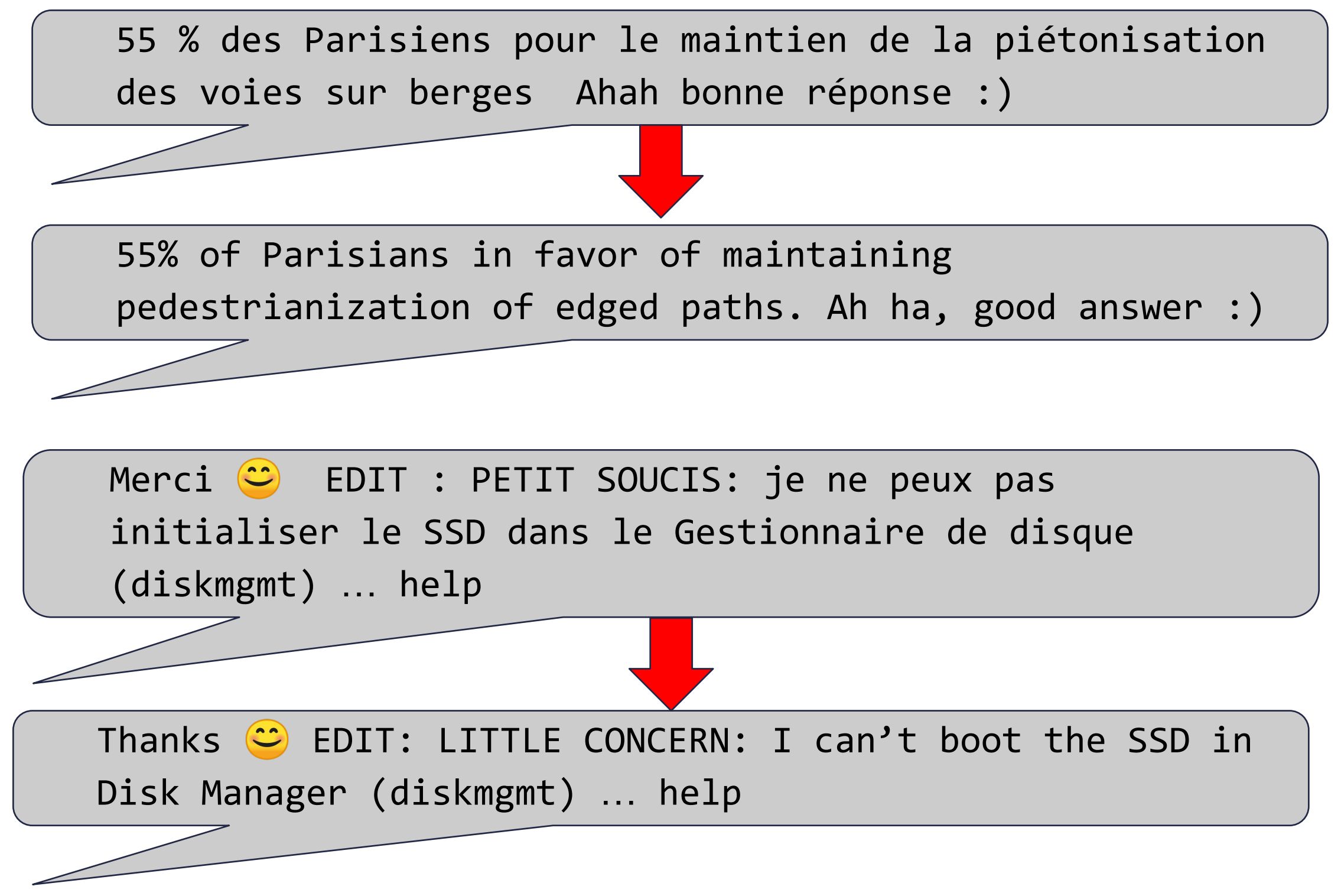
Introduction

Our goal is to improve on previous work on noisy machine translation. Our chosen paper presented a dataset of noisy comments from Reddit and their translation into English, French, and Japanese. The noise in the text data comes from several sources: *misspellings*, *poor grammar*, *idiomatic phrases*, and use of *emojis* and *emoticons*. All of these factors make translation more challenging.

The paper provides some results from a basic RNN machine translation model on their novel dataset. We wanted to make a model that is more state-of-the-art and more task-specific. We plan to train this model on a standard MT dataset as described in the paper, and on both the less noisy dataset and the noisy MTNT dataset, and compare performance.

Data

We use the MTNT dataset of Reddit comments in English translated into French and Japanese as well as French and Japanese comments translated into English. There are 7k - 37k sentences per language pair. We also use some standard MT datasets in order to compare to the baselines given in the paper, and to provide some pretraining. We also compare how the standard and refined model do on the standard datasets.



55 % des Parisiens pour le maintien de la piétonisation des voies sur berges Ahah bonne réponse :)

55% of Parisians in favor of maintaining pedestrianization of edged paths. Ah ha, good answer :)

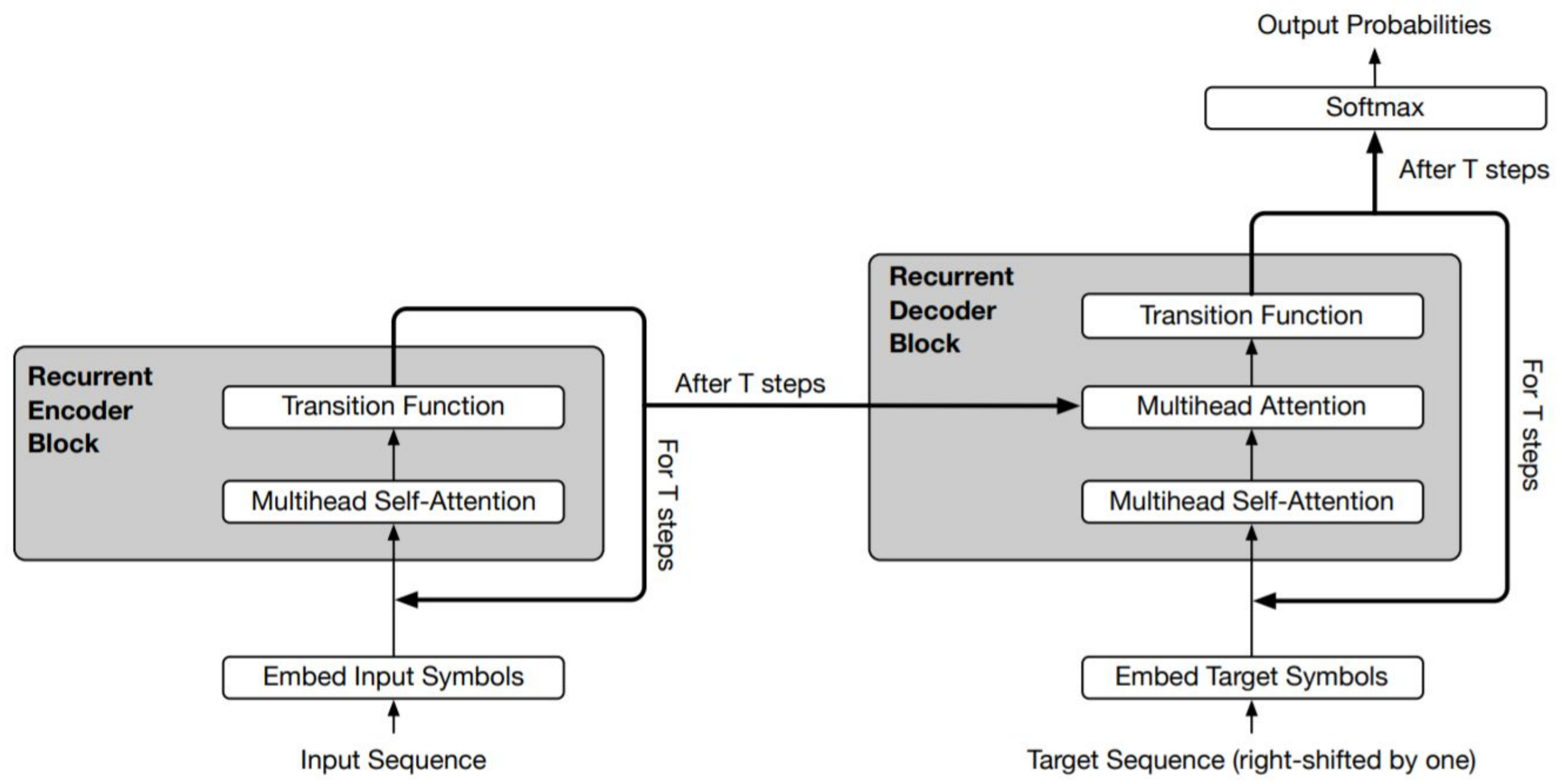
Merci 😊 EDIT : PETIT SOUCIS: je ne peux pas initialiser le SSD dans le Gestionnaire de disque (diskmgmt) … help

Thanks 😊 EDIT: LITTLE CONCERN: I can’t boot the SSD in Disk Manager (diskmgmt) … help

Note the atypical phrases and symbols used in typical Reddit comments.

Methodology

We use a Universal Transformer model for machine translation, which is a variation on the normal transformer model incorporating recurrent elements. Rather than recurring on ***positions*** in the sequence, the Universal Transformer recurs on ***revisions*** of its representations of each position in the sequence. This generalizes a Transformer with stacked encoder/decoder blocks, and is thought to better model some kinds of data. As the original Universal Transformer paper had code using Tensorflow, we implemented this model in Pytorch.



M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and Lukasz Kaiser. Universal transformers, 2018.

The MTNT website has a leaderboard of BLEU scores for how well their model performs in translating noisy phrases among English, French, and Japanese. A goal for our model was to produce scores comparable with what the authors of the original paper had. French/English BLEU accuracy averaged around 30% while English/Japanese scored around 11%.

From a practical view, we want our approach to noisy machine translation to not only perform well the MTNT dataset, but still function well on traditional machine translation datasets. To that end, we pretrained our model over several epochs on the Hansard French-English dataset.

Results

Compute Time:
We trained our model on a GCP virtual machine with 50 GB of RAM, 8 vCPU cores, and a P100 graphics card.

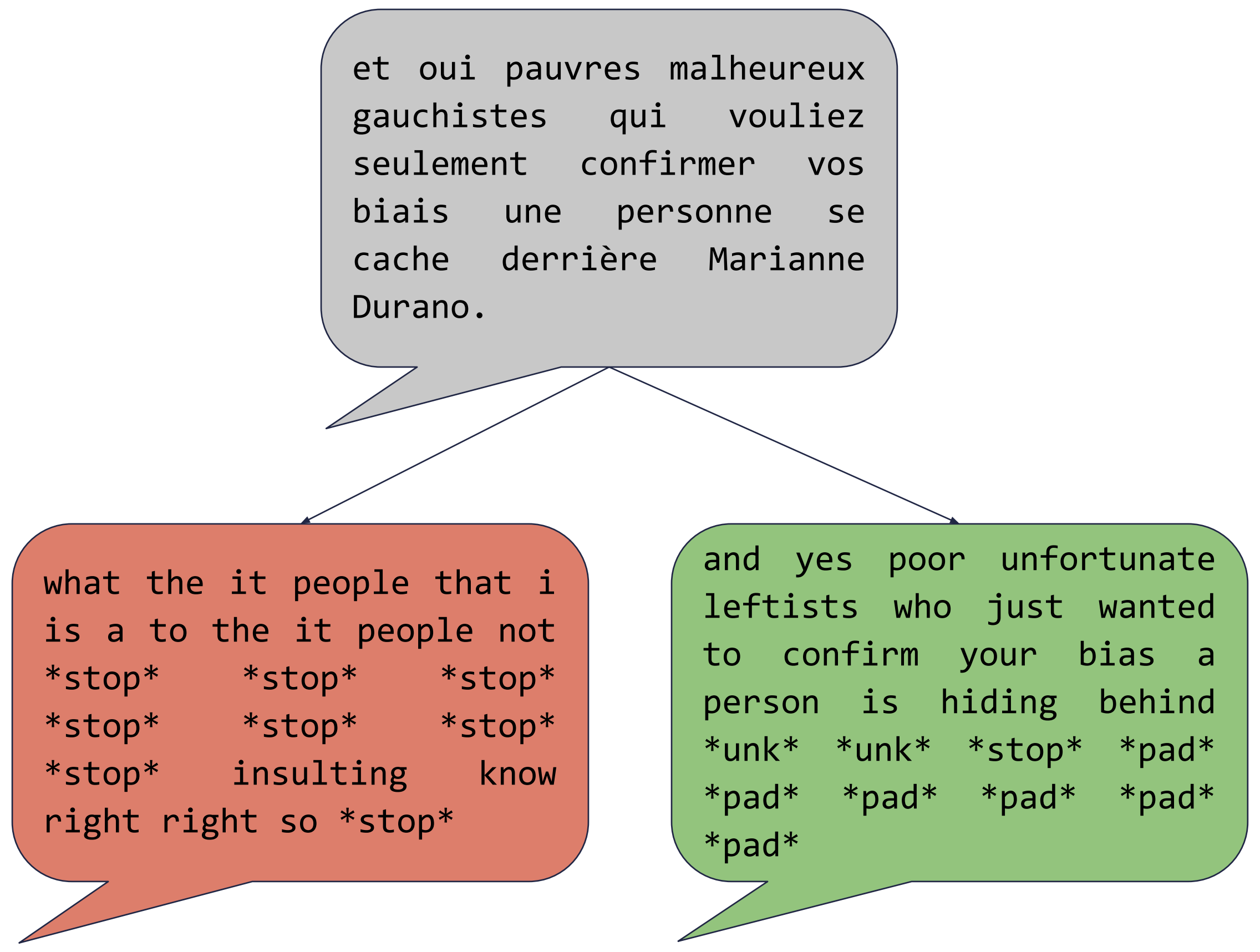
Our Universal Transformer uses the following hyperparameters, determined via testing:

Hyperparameter	Optimal Value
T	8
dropout	0.5
embedding size	64
hidden layer size	1024
attention heads	8
batch size	200

Below we report the quantitative results of our model on several training loops:

Perplexity, Accuracy, BLEU Scores

Training	Test on Hansard	Test on MTNT
5 epochs on Hansard	44.8, .3861	74764, .061, 0.1
20 epochs on MTNT	2615, 0.21	657, .145, 6.0
5 epochs on Hansard, 20 epochs on MTNT	14, .52, 8.8	380, .179, 6.9



et oui pauvres malheureux gauchistes qui vouliez seulement confirmer vos biais une personne se cache derrière Marianne Durano.

what the it people that i is a to the it people not *stop* *stop* *stop* *stop* *stop* *stop* *stop* insulting know right right so *stop*

and yes poor unfortunate leftists who just wanted to confirm your bias a person is hiding behind *unk* *unk* *stop* *pad* *pad* *pad* *pad* *pad* *pad*

An example of a less-than-ideal translation produced by the Universal Transformer.

Conclusion

We were able to reproduce a version of the Universal Transformer that exceeds the performance of a standard Transformer as implemented during the course on a standard MT dataset. However, it struggles to find effective translations of the extremely noisy MTNT dataset. We had expected to be able to more or less match the accuracy described in the paper, but we fell moderately below that threshold.

One conclusion we can draw is that pretraining on the Hansard data is beneficial. We were somewhat surprised by this, given the significantly different styles of the two datasets.

We believe that several factors contributed to the low performance of our model on the noisy data.

- High test perplexity compared to train perplexity on MTNT indicates overfitting, likely due to the small size of the dataset. This is despite our use of dropout layers in both the transformer blocks and dense layers.
- While we tuned hyperparameters to some extent, we lacked the time and compute resources to do a rigorous hyperparameter search.
- We weren’t able to preprocess MTNT effectively since standard tokenization isn’t suitable for emoticons and Internet slang.

¯\_(ツ)_/¯ → ¯\_(ツ)_/¯

Acknowledgements

We would like to acknowledge our mentor TA James Wang for his help with the project.

The compute resources were obtained from credit supplied to CSCI 1470 by Google.