STAP: Simultaneous Translation And Paraphrase

Dmitry Grigorev, DS2

Vakhitova Alsu, DS1

Ilalova Alina, DS2

Kozlov Pavel, RO

Abstract

This report describes the goals of the project, the methods that were used in implementation and final results with performance scores of the model.

Introduction

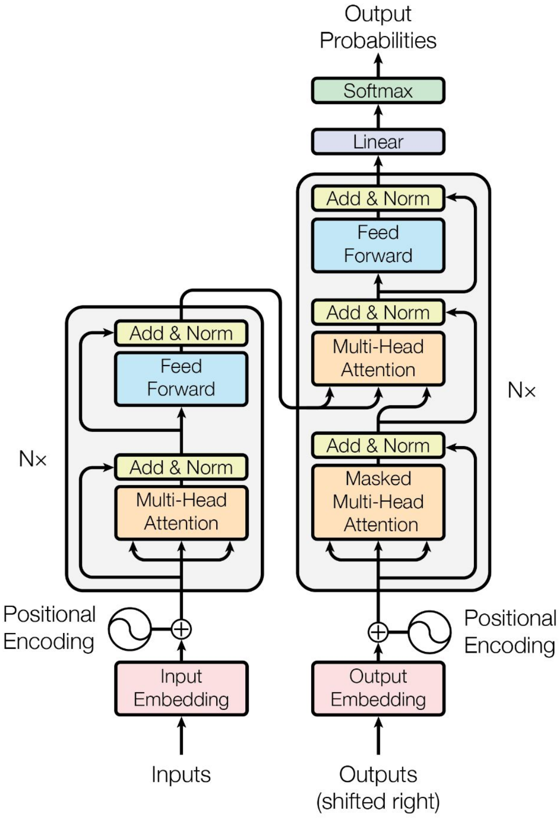
Machine translation systems typically produce a single output, but in certain cases, it is desirable to have many possible translations of a given input text. This situation is common with language-learning platforms, where some learning happens via translation-based exercises, and grading is done by comparing learners' responses against a large set of human-curated acceptable translations. We believe the processes of grading and/or manual curation could be vastly improved with richer, multi-output translation and paraphrase systems.

A portion of learning on language-learning platforms happens through translation-based exercises. In this project, we focus on the challenges where users are given a prompt in the language they are learning (Russian), and type a response in their native language (English). The developed system will provide the opportunity for students to learn the language in a more flexible way.

The most efficient translation systems nowadays use transformer-based language models as a core (Google translate, Yandex etc.). However, “pure” transformer models have seq2seq architecture, so they are able to produce only one variation of the correct translation. To overcome it some additional features are used which were researched as a part of our work.

Model description

Our final model consists of consequent translation and paraphrase models.
Neural Machine Translation requires a lot of computational resources in order for a model to produce adequate results, so we decided to use pre-trained FairSeq model by Facebook AI Research. This model was developed as a WMT’19 news translation task submission and won it in four tasks: English to German, German to English, English to Russian and Russian to English.



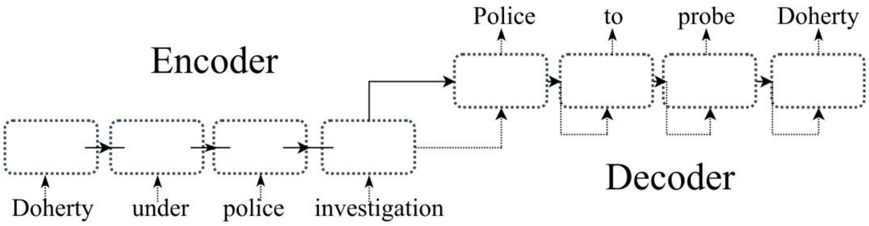
The model is based on big Transformer architecture by [Vaswani et al., 2017]. The Transformer uses stacked self-attention and point-wise, fully connected layers for both the encoder and decoder. The representation of the model architecture is shown on the right side of the page.

The core new idea of the translation model was to use a backward translation mechanism. It can be described as follows:

- first a forward model produces several candidates for translation from source language to target language
- then a backward model produces translation from target language to source language for each candidate
- then a language model compares the results of backward model with original inputs
- finally, the system selects the hypothesis with the highest combined score according to all models as the actual translation

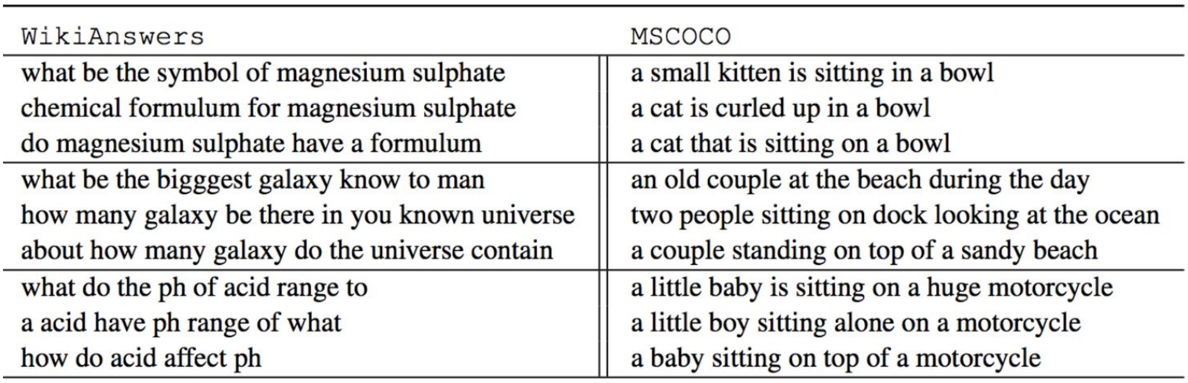
The authors also used more advanced technique in dataset cleaning such as removing instances where a translation is significantly longer than its corresponding translation and language identification filtering to keep only those sentence pairs with the correct languages on both sides.

The paraphrase model consists of Encoder-Decoder model, Residual LSTM network and we also added residual connections between LSTM layers. When stacking multiple layers of neurons, the network often suffers through a degradation problem. Residual connections can help overcome this issue. This allows for efficient training of deep LSTMs. The schema of our model is below:



Datasets

As it was mentioned earlier, pretrained model was used for translation task, so we did not have to use any dataset for translation. However, we have used 2 datasets for paraphrase: MSCoco and WikiAnswers. Samples of sentences for both of these datasets can be seen below:



WikiAnswers	MSCOCO
what be the symbol of magnesium sulphate chemical formulum for magnesium sulphate do magnesium sulphate have a formulum	a small kitten is sitting in a bowl a cat is curled up in a bowl a cat that is sitting on a bowl
what be the bigggest galaxy know to man how many galaxy be there in you known universe about how many galaxy do the universe contain	an old couple at the beach during the day two people sitting on dock looking at the ocean a couple standing on top of a sandy beach
what do the ph of acid range to a acid have ph range of what how do acid affect ph	a little baby is sitting on a huge motorcycle a little boy sitting alone on a motorcycle a baby sitting on top of a motorcycle

MSCoco is a large-scale object detection, segmentation, and captioning dataset. COCO has several features, but we have used the text representation of captions for images. The number of captions in this dataset is 413,915 captions for 82,783 images in training, 202,520 captions for 40,504 images in validation and 379,249 captions for 40,775 images in testing.

The WikiAnswers corpus contains clusters of questions tagged by WikiAnswers users as paraphrases. Each cluster optionally contains an answer provided by

WikiAnswers users. There are 30,370,994 clusters containing an average of 25 questions per cluster. 3,386,256 (11%) of the clusters have an answer.

Implementation

The project was implemented using Python 3.6.9. Python is an interpreted, high-level, general-purpose programming language. It is extremely popular in scientific community. Python's greatest strength is its huge number of libraries which can help one do a variety of tasks, such as web services, multimedia, databases, text, and image processing. The reason for usage of Python version, which was released on the second of July 2019, is that it is more stable than Python 3.7 and all libraries are adapted for it (which is not the same for version 3.7).

The translation model was written on Pytorch, so we used it as a deep learning framework and we also used TorchHub as a model source.
The paraphrase model was implemented using Tensorflow framework. TensorFlow is an open source library for numerical computation, which is generally heavily used for machine learning applications such as neural networks. TensorFlow has comprehensive, flexible ecosystem of tools, libraries and community resources which makes the library on of the standards for computer science researchers.

Results

Translation model scored with 40.0 points for BLEU metric.
On the other hand, the evaluation of the paraphrase model yielded following scores: BLEU=20.3, METEOR=23.1.

Application Production Techniques

In addition to model implementation we also did not forget about product development techniques and tried to follow processes that are similar to industrial ones.

Continuous Integration pipeline

In order to be sure that our code is always working and performs at the expected level, the template of CI pipeline was created. It consists of 2 stages: Build and Test. The Integration and Deploy stages are missed because the deployment to any “production” server is not expected for this project.

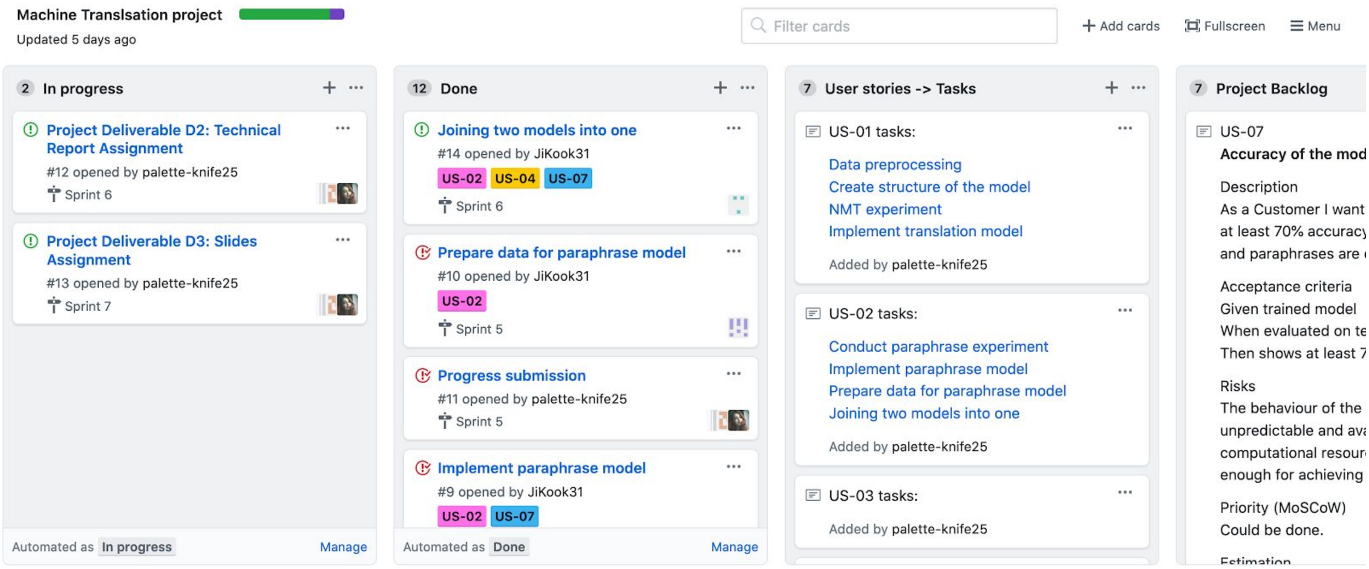
The idea of the pipeline is simple: during the Build stage the dataset is loaded and the model is created, trained and saved to artefacts. At the Test stage the model is loaded from the artefact that was saved in the previous stage and evaluated using test set. There are only tests that are performed at the moment: checking that the model file is successfully loaded and checking that accuracy score of the model is higher than 75%. In case if any of the test fails, the whole pipeline fails and appropriate message is shown to the developer.

At the end, the final model file from artefacts can be used for predictions. The GitHub pipeline can be found by this link.

Scrum Activities

Due to the reason that this project required weekly progress submissions, we decided to follow Scrum processes in order to be able to work on the project slowly during the week and to avoid situations with “overnight” implementation of the project.

Kanban board was the most useful for our team, because every task could be viewed in a very convenient way and everyone was aware of what each team member is working on right now. It was also very helpful in planning the next activities and tasks, because everything can be written on the board and no little aspect will be forgotten. For the board we used the GitHub Project Board, as all of our source code is stored in GitHub VCS and we decided that to use the same system will be more convenient way than to switch between tools back and forth. The screenshot of our board can be seen below. As it can be noticed, we also have User Stories and they all are linked to tasks in order for us to know which user story is already implemented and which needs more attention to it.



Machine Translsation project
Updated 5 days ago

2 In progress

- Project Deliverable D2: Technical Report Assignment
  #12 opened by palette-knife25
  Sprint 6
- Project Deliverable D3: Slides Assignment
  #13 opened by palette-knife25
  Sprint 7

Automated as In progress — Manage

12 Done

- Joining two models into one
  #14 opened by JiKook31
  US-02 US-04 US-07
  Sprint 6
- Prepare data for paraphrase model
  #10 opened by JiKook31
  US-02
  Sprint 5
- Progress submission
  #11 opened by palette-knife25
  Sprint 5
- Implement paraphrase model
  #9 opened by JiKook31
  US-02 US-07

Automated as Done — Manage

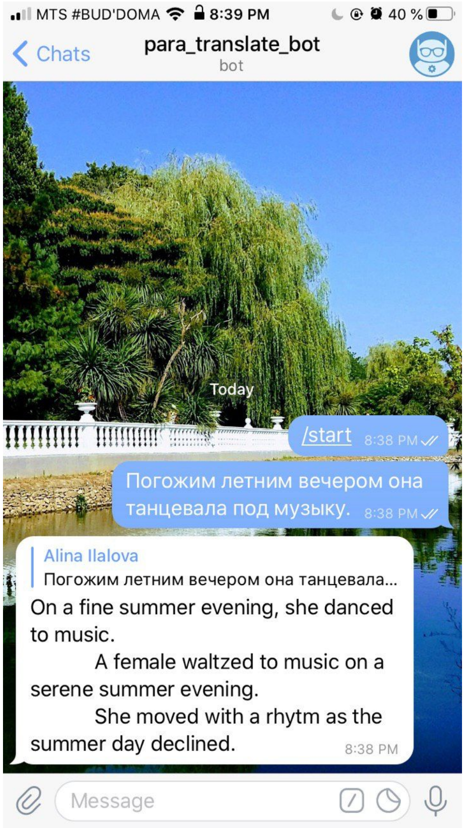
7 User stories -> Tasks

- US-01 tasks:
  Data preprocessing
  Create structure of the model
  NMT experiment
  Implement translation model
  Added by palette-knife25
- US-02 tasks:
  Conduct paraphrase experiment
  Implement paraphrase model
  Prepare data for paraphrase model
  Joining two models into one
  Added by palette-knife25
- US-03 tasks:
  Added by palette-knife25

7 Project Backlog

- US-07
  Accuracy of the mod
  Description
  As a Customer I want
  at least 70% accuracy
  and paraphrases are
  Acceptance criteria
  Given trained model
  When evaluated on te
  Then shows at least 7
  Risks
  The behaviour of the
  unpredictable and ava
  computational resour
  enough for achieving
  Priority (MoSCoW)
  Could be done.
  Estimation

Telegram Bot



/start 8:38 PM

Погожим летним вечером она танцевала под музыку. 8:38 PM

Alina Ilalova
Погожим летним вечером она танцевала...
On a fine summer evening, she danced to music.
A female waltzed to music on a serene summer evening.
She moved with a rhytm as the summer day declined. 8:38 PM

In addition to implementing the model for translation and paraphrase, we decided to create a Telegram Bot as well. It will provide a convenient way for users to interact with our created model and help them in their daily tasks. It works very in a very simple way: you just start it and insert a sentence or word that you would like to get several translations for and it will send you an answer with as many possible words or sentences as the model could generate.
The screenshot of the bot can be seen in the right side of the page.

Discussion and Future Work

During implementation of this project we faced some issues, and one of them is low quality of paraphrase datasets. We were not able to find any dataset that would satisfy us, so we had to use the ones that were available. In addition to that, there were no suitable dataset for combined translation and paraphrase.

Moreover, computational resources are always the issue in Machine Learning and this project is not an exception. We would be able to produce better results if we had access to more powerful machines. Nevertheless, we managed to achieve satisfactory results even with our current set of computational resources.

We also have a few ideas on how to improve our project. First is to create one model that performs both tasks: translation and paraphrase, without any division and task-specific models. We could not achieve this goal due to absence of good quality dataset, as it was mentioned earlier and lack of computational resources.

Another possible improvement is to implement mobile application instead of Telegram bot, because it will be even more convenient for users. Unfortunately, implementing mobile application goes way beyond this project so we were not able to do that.

Conclusion

In this project we have stated the problem of one-to-one translation and proposed our solution to it. Furthermore, we have described details of the architecture and implementation. The results measured with classical sequence-to-sequence metrics were presented. Moreover, we have reflected on teamwork techniques and modern practices that were utilized. Finally, some speculations on possible future improvements were posed.

References

https://github.com/JiKook31/DL_project
Facebook FAIR's WMT19 News Translation Task Submission