Project Report

Moldir Koishybayeva

`moldir.koishybayeva@univaq.it`

University of L'Aquila — January 26, 2023

Introduction

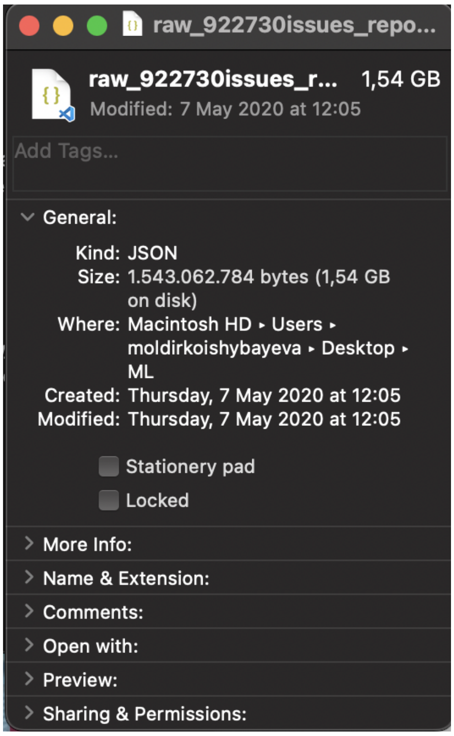
Project's purpose is to develop a deep learning tool for summarizing long text to retrieve a short description. The project involves generating text descriptions for GitHub issues. Three models of the Transformer library presented by HuggingFace, such as BERT, BART, and T5, have been applied. The dataset provided by the iTiger project, which is in public access, was used for training purposes. The obtained results have been evaluated. Project's Git Repository:
`https://github.com/m6search/summarization-of-github-issues`

ℹ

> **Info:** Objective is to employ different Natural Language Processing Techniques to generate a title of GitHub issue based on the following research work.
> For more detailed information: iTiger: An Automatic Issue Title Generation Tool (Zhang et al., 2022)

1 Dataset

Provided dataset consists of 922730 entries, and a total of 4 columns, namely number, title, body, and repository. (Zhang et al., 2022)



raw_922730issues_repo...

raw_922730issues_r... 1,54 GB
Modified: 7 May 2020 at 12:05

Add Tags...

General:

Kind: JSON
Size: 1.543.062.784 bytes (1,54 GB on disk)
Where: Macintosh HD ▸ Users ▸ moldirkoishybayeva ▸ Desktop ▸ ML
Created: Thursday, 7 May 2020 at 12:05
Modified: Thursday, 7 May 2020 at 12:05

□ Stationery pad
□ Locked

Fig 1. Dataset

2 Data Preprocessing

During preprocessing, data[‘number’] column was removed, since it is not essential for the summarization of the text in the title generation. Further steps were performed such as:

1. data[‘body’] column
   - replacing code snippets with word ‘code’
   - replacing added image link with word ‘image’
   - replacing hyperlink text with word ‘link’
   - replacing url text with word ‘url’
   - removing new lines
   - lowering text
2. data[‘title’] column
   - removing tags
   - removing emphasis
   - lowering text

3 Exploratory Data Analysis

From the illustration below, we can clearly observe from *almost 1 million GitHub issues* that coders usually mention words such as *support, work, fail, working, error, using, add, window, value*, while providing a title for the issue.

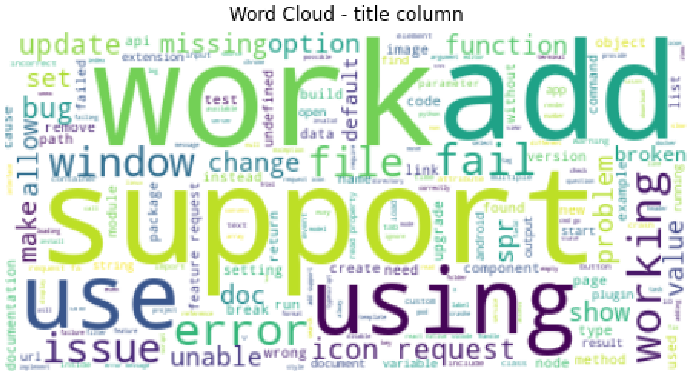


Fig 2. Word Cloud. Title column

From Figure 4 it can be noted that the most common repositories are *microsoft/vscode* (77796 issues), *golang/go* (34730 issues), and *kubernetes/kubernetes* (32704 issues).

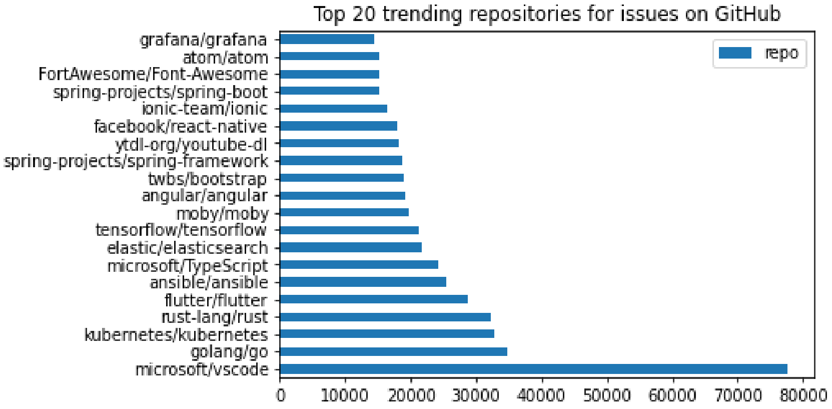


Fig 3. Top 20 trending repositories for issues on GitHub

After understanding our dataset, let's now proceed to model training and their evaluation.

4 Modelling

HuggingFace's Transformer provides pre-trained models for natural language processing purposes. Utilizing a pre-trained model for summarization purposes includes these steps:

- Data Preparation

- Loading a pre-trained model
- Fine-tuning the model
- Model evaluation

Functions that are provided by the Hugging Face’s transformers library were included while implementation of the **iTiger project**.

- **AutoConfig** is an automated tuning of models based on transformers. Using AutoConfig, it can optimize the hyper-parameters of a transformer model.
- **AutoModelForSeq2SeqLM** is a function for automatically selecting and configuring models.
- **AutoTokenizer** is a function for automatically selecting tokenizer for a specified model
- **Seq2SeqTrainer, Seq2SeqTrainingArguments** are imported for the purpose of training sequence-sequence models.
- **MBartTokenizer** provides tokenization for the MBart model.
- **MBartTokenizerFast** provides tokenization for the MBart model.

For the models presented (BART, BERT, and T5), the `num_beams` parameter was set to 1, `max_length = 20`, and `min_length=10` to generate shorter text for the issue title. If the `num_beams` parameter is higher, it will generate more diversified text. Parameters `min_length` and `max_length` represent the minimum and maximum length of the generated text, respectively.

4.1 BART

Command Line

```
model= "sshleifer/distilbart-cnn-6-6"
bart_tokenizer = AutoTokenizer.from_pretrained(model)
bart_model = BartForConditionalGeneration.from_pretrained(model)
```

distilbart-cnn-6-6 is a fast version of the cnn-based BART model. *AutoTokenizer* is a tokenizer that automatically tokenizes with appropriate tokenization method. *BartForConditionalGeneration* is a class for fine tuning the DistilBART model (AWS, n.d.).

4.2 BERT

BERT model training is computationally intensive, involving a robust GPU and a considerable volume of data.

Command Line

```
tokenizer = AutoTokenizer.from_pretrained("bert-base-cased")
model = BertForMaskedLM.from_pretrained("bert-base-cased")
```

BertForMaskedLM is a class for fine-tuning the BERT model to the Masked Language Modeling task.Tunstall, L., Werra, L.von and Wolf, T., 2022

4.3 T5

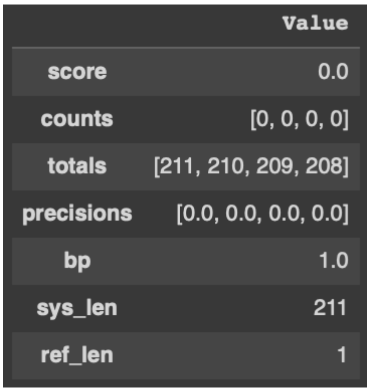
Command Line

```
model = T5ForConditionalGeneration.from_pretrained('t5-small')
tokenizer = T5Tokenizer.from_pretrained('t5-small')
```

T5-small is the text-to-text transformer model version. It is a large-scale pre-trained transformer model developed by Google Research.

5 Evaluation

BLEU (Bilingual Evaluation Understudy) is a text quality evaluation method. This method compares the generated text to one or more reference texts and determines a rating.



	Value
score	0.0
counts	[0, 0, 0, 0]
totals	[211, 210, 209, 208]
precisions	[0.0, 0.0, 0.0, 0.0]
bp	1.0
sys_len	211
ref_len	1

Fig 4. Evaluation of BERT with BLEU

According to the Figure 5, we can recognize that the BART model is more applicable for paraphrasing larger text.

It can be stated that the BART, T5 models provide shorter generated title. At the same time, the accuracy is poor. It improves as the length of the text increases.

The results can be improved by training the model on a smaller dataset. It should be noted that this requires a decent amount of RAM, and the capabilities of google colab do not provide it.

References

AWS, A. (n.d.). *Set up a text summarization project with Hugging Face Transformers: Part 2*. Available from: `https://aws.amazon.com/blogs/machine-learning/part-2-set-up-a-text-summarization-project-with-hugging-face-transformers/`.

Tunstall, L., Werra, L.von and Wolf, T. (2022). *Natural language processing with transformers: Building language applications with hugging face*. O Reilly Media.

Zhang, T., Irsan, I. C., Thung, F., Han, D., Lo, D., and Jiang, L. (Nov. 2022). “iTiger: an automatic issue title generation tool”. In: *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. ACM. DOI: `10.1145/3540250.3558934`. Available from: `https://doi.org/10.1145%2F3540250.3558934`.