# BoloBot: a tourist helper chatbot
# NLP Course Project & Project Work

**Marco Acerbis, Xiaotian Fan**
Master's Degree in Artificial Intelligence, University of Bologna
{ marco.acerbis, xiaotian.fan }@studio.unibo.it

## Abstract

In this work we present our study aimed at the creation of helper chatbot for tourists visiting the city of Bologna. After defining the main modules and adopted solution, we focused on evaluating which transformer model implement for each task. In particular, we fine-tuned the T5-small model on two different tasks, summarization and translation, in order to provide to the chatbot the tool to generate answers by summarizing relevant Wikipedia articles retrieved by a Semantich Search algorithm, and also provide support in different languages like Italian and English.

## 1 Introduction

Our work aims is to introduce a way to help tourists visiting the city of Bologna, or any other city, by providing a simple tool that gives insights about the city itself, the local culture and other information that can help tourists enjoying their stay. Given these aspects of our work, we needed also to account for the possibility of non-English speakers visitors, thus we decided to implement a translation module to allow the final user to ask questions in a different language, like Chinese.
Our system, describe in details in Sec.3, is divided into two main modules:

- Question-Answering (QA) Module that utilizes **Semantic Search**, to recover relevant pieces of information from *Wikipedia* articles, and a fine-tuned model to summarize the contents into an answer;
- A translation module that implements a fine-tuned model for translations that supports three languages: English, Italian and Chinese.

In particular, we opted for this solution for QA to overcome two main problems with this type of algorithms. First, during development, we realized that it is harder to train an efficient QA model than a model focused on summarizing articles. On the other hand, this approach also allow for a lighter implementation of the final chatbot given that semantic search algorithms are less computationally expensive than LLMs.
In developing our system, we decided to study and develop each part alone and only at the end we put the obtained results together to create the final chatbot. As described in Sec.5, we started by evaluating different semantic search algorithms in order to select the best performing one to use in our final implementation. After that we focused on re-training our summarizer on article-like documents, carrying out multiple test/evaluation seed runs and selecting the best performing model between BERT and T5. At last, for the project work extension, we fine-tuned BERT-Multilingual to complete translation CH-EN and EN-CH, evaluating the performance both qualitatively and quantitatively, in each case.

## 2 Background

We are developing a multilingual QA chatbot project specifically designed for tourists visiting Bologna. This project also aims to address the language barrier faced by tourists from different countries and cultural backgrounds. The bot will support three main languages: English, Italian and Chinese. By providing multilingual support, our goal is to ensure that tourists can easily access the necessary travel information, resolve queries, and enjoy their journey.

The key features of our multilingual QA chatbot include:

1. Travel Information: The bot will provide detailed information about Bologna's tourist attractions, cultural heritage, cuisine, activities, and events;
2. Frequently asked questions: The bot will answer common questions related to weather

conditions, local customs, currency exchange, emergency contact information, and other relevant topics to help visitors better adapt to the local environment;

3. Language support: Users can choose to communicate with the bot in English, Italian, or Chinese. The bot will understand and generate text in the respective languages.

## 3 System description

Our system is divided into two modules that are then used by our chatbot: a Question-Answering module and a Translation Module.

### 3.1 Question-Answering (QA) Module

The main goal of our chatbot is to provide information about the city and the local culture to tourists, thus we implemented a QA module that provides answers in the form of summaries of Wikipedia articles.
The implementation is based on Semantic Search, in order to recover the articles with the relevant information, fed to a fine-tuned model that provides the answers by summarizing the retrieved documents. Note that both models has been re-trained on English texts samples.

#### 3.1.1 Semantic Search

Semantic Search consist in a mechanism to retrieve pieces of information based on the content of a given query. Differently from classical search based on keywords, this method allows us to "search with meaning", allowing us to exploit not only some relevant words but the entire query structure, by creating the relative embeddings, to find the desired documents. We decided to implement this solution because it resulted computationally cheap, and therefore suitable for mobile application like our chatbot.
In order to implement the most efficient model, we evaluated the performances of three different sentence transformers: *all-datasets-v3-mpnet-base*, *multi-qa-mpnet-base-dot-v1* and *bert-base-nli-mean-tokens*. The experiment and the custom metrics we used to evaluate the performances of the models are described in sec.5.
Also, the search algorithm covers an other important role and can provide some important benefits in reducing the time needed to recover the documents. While we briefly tested different search algorithms, we decided to run the experiment using FAISS (Facebook AI Similarity Search) score for efficient search, being one of the best performing solutions. On the other hand, the chatbot makes use of **Pinecone** to store the embedded dataset, a service that utilizes a custom search algorithm.

#### 3.1.2 Summarization

Instead of generating answers, we opted for summarizing the content of relevant documents provided by the Semantic Search algorithm. For this task, we followed the Seq2Seq approach and fine-tuned both BERT-base and T5 models to perform summarization. In particular, we re-trained the models on the **CNN-Dailymail** dataset composed of news articles and their summary. The experimental set-up and the obtained results are reported in Sec. 5.
In the final application, once the chatbot fetches pertinent articles from Picone, it forwards the text documents to our fine-tuned model, which then generates a summary for the user's query. This summary is subsequently presented to the user as an answer.

### 3.2 Project Work: Translation Module

For our project work, we decided to implement a translator module to allow the user to input queries in three different languages. Instead of fine-tuning each of the previous models on each language (English, Italian, Chinese), we found optimal to only train those model on English data, that is largely available and ready for different tasks, and to used the translated text as input when the user's query is an a different language. We initially wanted evaluated the performances of BERT-Multilingual, especially in the translation task from Chinese to English and vice versa. Unfortunately, we did not get good results, as described in Sec.5, and we opted to test fine-tuning the T5-small for the translation task, focusing in particular on the English-Italian conversion(s).

## 4 Data

We used the following datasets to fine-tune our models or to create our custom dataset of information about the city of Bologna:

- **Wikipedia Snippets**: a large collection of Wikipedia articles with page and section titles. We used this dataset to evaluate the different models for Semantic Search and also to create

our custom dataset with articles about the city of Bologna;

- **CNN-Dailymail 3.0.0**(Grusky et al., 2018): a dataset of about 300 000 newspaper articles with their summary. We used this dataset to fine-tune BERT on the summarization task;
- **WMT18 zh-en**: a large dataset of Chinese-to-English translations texts samples. We employed this dataset to fine-tune the bert-base-multilingual-cased model in our study of the traslator module:
- **Opus-books**: a collection of passages from copyright-free books in different languages. We adopted this dataset to fine T5-small on the translation task involving Italian and English texts.

## 5 Experimental setup and results

We run our experiments and test on Google Colab due to the requiement of high-performing GPUs, even if some time and resource constrainets forced us to modify and adopt different aspects of our project.
First of all, we tried to reproduce simple implementations of the different modules. Once we learnt to handle the different libraries and models, we started adapting them for our chatbot application.

### 5.1 Finding the optimal model for Semantic Search

We started or analysis by evaluating three different models for Semantic Search that we selected from examples provided by Huggin Face or other literature sources. In particular, we studied *all-datasets-v3-mpnet-base*, *multi-qa-mpnet-base-dot-v1* and *bert-base-nli-mean-tokens*.
In order to evaluate which model performed the best, we carried out the following experiment:

1. We extracted 1000 sample articles with their titles from the Wikipedia Snipped dataset;
2. Each model has been tested in the following way: provided the title of the article the model had to retrieve the correct article from the embedded dataset in the Top 2 outputs;
3. In order to evaluate the performance of each model we defined our custom metric: if the correct document was the best match (Top 1) we assigned 10 points, if it was the second best guess we assigned 5 points and we assigned 0 points if the correct document was not present in the Top 2 retrieved documents.

As we can see in Tab.1, all-datasets-v3-mpnet-base and multi-qa-mpnet-base-dot-v1 models obtained similar results with the second one performing slightly better than the former (9170 points and 8995 points respectively); thus making it our choice for implementation in the final chatbot application. On the other hand the last model we tested, bert-base-nli-mean-tokens, performed poorly with a score of only 1335 points out of 10000.

| Model | Score |
|---|---|
| all-datasets-v3-mpnet-base | 8995 |
| multi-qa-mpnet-base-dot-v1 | 9170 |
| bert-base-nli-mean-tokens | 1335 |

Table 1: Semantic Search algorithm evaluation results

Once we selected the model for Semantic Search, we proceeded by creating the embeddings index on Pinecone: while this operation is simple to perform it requires a lot of time. That happens because, in order to avoid downloading excessively large datasets, we utilized the *"streaming"* option for the datasets available through Hugging Face. On the other hand, this slowed down the operation because documents are made available on request and checked one-by-one until the desired number is collected. Fortunately, this operation must be processed only one time as Pinecone offers the possibility to store the generated index for free as long it is regularly used.

| Model | Seed | Val Precision | Val Recall | Val F1 |
|---|---|---|---|---|
| bert2bert | 25 | 0.0005 | 0.0079 | 0.0010 |
| bert2bert | 42 | 0.0001 | 0.0016 | 0.0002 |
| T5-small | 25 | 0.1117 | 0.2155 | 0.1436 |
| T5-small | 42 | 0.0689 | 0.2075 | 0.1025 |
| **Model** | **Seed** | **Test Precision** | **Test Recall** | **Test F1** |
| bert2bert | 25 | 0.0015 | 0.0027 | 0.0019 |
| bert2bert | 42 | 0.0008 | 0.0022 | 0.0012 |
| T5-small | 25 | 0.0761 | 0.1874 | 0.1064 |
| T5-small | 42 | 0.0577 | 0.1879 | 0.0871 |

Table 2: Fine-tuned summarizer models performances on validation and test sets evaluated with Rouge-2 metrics.

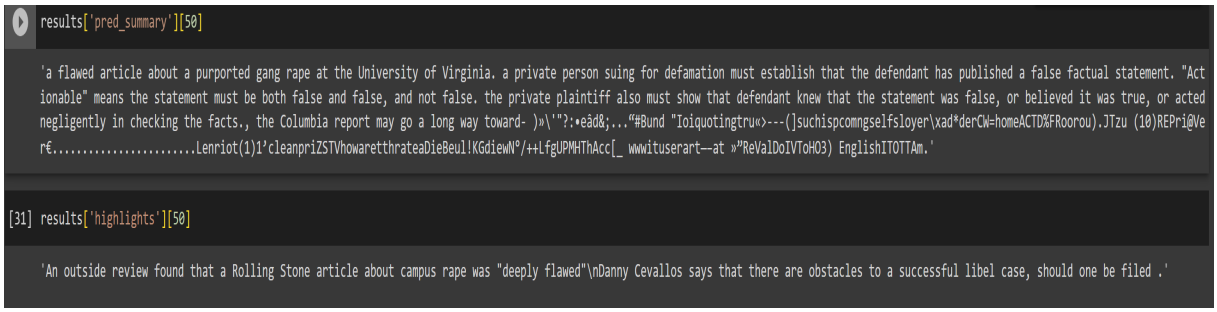


Figure 1: Example of summary generated by T5-small, fine-tuned on CNN-Dailymail dataset, compared with the "highlight" provided as "ground truth".

### 5.2 Fine-tuning the summarizer

To complete the first module of our chatbot, we needed to developed a model able to summarize the content retrieved by the Semantic Search. In particular, we experimented with two encoder-decoder models: bert2bert (Devlin et al., 2018) and T5-small (Raffel et al., 2019).
The main challenge we faced with this task was being able to obtain good results given the limitations of Google Colab (in general we faced a lot of OOM errors due to the large quantity of data needed to accurately re-train these models), therefore to evaluate which model implement in our application, we performed all the fine-tuning using only a limited portion of the data.
For each model, we performed two training/evaluation runs, each consisting of 3 epochs, with two different training seeds; while the performances have been evaluated using the **Rouge-2** metrics.

| Model | Val Precision | Val Recall | Val F1 | Test Precision | Test Recall | Test F1 |
|---|---|---|---|---|---|---|
| T5-small | 0.0427 | 0.1767 | 0.0679 | 0.1094 | 0.1436 | 0.1215 |

Table 3: Results of the new fine-tuning performed on T5-small to obtain better results.

From the results showed in Tab.2, we can clearly see that T5-small outperformed bert2bert in the summarization task, with scores that are about 100 times better both on validation and test sets. Also, BERT max length for inputs is limited to 512, while T5-small can accepts longer inputs.
A qualitative analysis of the texts generated with T5-small, like the one showed in Fig.1, showed that they are, in general, a good summary of the article, with many parts in common with the provided control label ("highlights"), especially in the first part of the sentence. Unfortunately, near the end of the sentence the generated text usually looks like a weird mix-up of random characters and words. For these reasons, we decided to conduct a new fine-tuning with more data in order to improve the generated summaries. Here a short list of the parameters we used for the new training and evaluation run:

- Encoder max length: 1024;
- Max length of the generated summary: 64, we tried to cut out the errors from the generated text;
- Number of samples in the training set: 1000;
- Number of samples in the validation set: 100;
- Number of samples in the test set: 500;
- Number of epochs: 3.

The results reported in Tab.3 shows that while we obtained slightly worse performances on the validation set, we managed to improve the final results on the test set by limiting the length of the generated outputs.

| Model | Task | Val Bleu | Test Bleu |
|---|---|---|---|
| bert-base-multilingual-cased | ZH-EN | 0.004900 | 0.0 |
| bert-base-multilingual-cased | ZH-EN | 0.004800 | 0.0 |
| bert-base-multilingual-cased | ZH-EN | 0.004600 | 0.0 |
| bert-base-multilingual-cased | EN-ZH | 0.002900 | 0.0 |
| bert-base-multilingual-cased | EN-ZH | 0.001800 | 0.0 |
| bert-base-multilingual-cased | EN-ZH | 0.004000 | 0.0 |

Table 4: SacreBleu scores after fine-tuning bert-base-multilingual-cased for translation task.

### 5.3 Fine-tuning the translator

As an extension of our chatbot, we experimented with a translator module, and in particular we studied two different encoder-decoder models: BERT-Multilingual and T5-small.

During this task, we encountered certain challenges correlated to the constraints imposed

by Google Colab that forced us to adop different strategies in order to obtain the desired, or at least qualitatively good, results. We started working with the MWT18 dataset to adapt BERT-Multilingual on the translation task involving Chinese and English texts. Working with the entire dataset, which comprises millions of sentences in both Chinese and English, along with BERT-Multilingual resulted impossible from the moment we kept reaching the limitations described before. Consequently, in order to avoid extended training times and potential GPU breakdowns, we had to work with a smaller dataset(s) for this task.

From the initial experiments involving translation between Chinese and English, as well as English to Chinese, the outcomes in Table 4 indicate considerably low values for the BLEU scores. Notably, all Test BLEU scores were 0. Evidently, these outcomes reveal that fine-tuning of Bert-base-multilingual-cased isn't effective with a limited dataset. Our training set included 1200 examples, while the validation and test sets had 500 and 300 examples respectively. The maximum length for both the Encoder and Decoder was set at 512, and each train run spanned 3 epochs.

After the bad performance we observed with BERT-multilingual, we decided to try fine-tuning T5-small again, but this time on a different task. We also decided to change the languages under analysis for the translation task: the model has been re-trained to perform translations from Italian to English and vice versa. As for the other trainings, we set-up multiple training/evaluation runs with 2 epochs each (to avoid getting timed out by Colab). We collected the sample data from the opus-books dataset, and in particular we used 20000 samples from the train split as training set and 5000 samples for the validation set; while the test was made of 500 samples from the original test split of the dataset. As for the metrics, we decided to keep evaluating the model's performances with **SacreBleu**.

The performances of our translator are reported in Tab.5. In details, we observed that the fine-tuned model performed very well when had to translate from Italian to English. On the other hand, it showed poor results in the inverse task of translating from English to Italian.

| Model | Task | Seed | Val Bleu | Test Bleu |
|---|---|---|---|---|
| T5-small | IT-EN | 25 | 0.7904 | 0.7687 |
| T5-small | IT-EN | 42 | 0.8044 | 0.7793 |
| T5-small | EN-IT | 25 | 0.2619 | 0.1856 |
| T5-small | EN-IT | 42 | 0.1723 | 0.0911 |

Table 5: SacreBleu scores after fine-tuning T5-small for translation task.

## 6 Discussion

The results of our experiments in the different tasks showed that the main challenge to overcome when working with LLMs, like BERT or T5, requires not only huge amount of data, but also powerful machines able to handle both the data and the models.
As show by our experiments on Semantic Search, there are some alternative solutions to pure LLMs that allows for practical and at the same time functional solution that are computationally cheaper to run.
We are also satisfied by the result obtained by our fine-tuned summarizer that we obtained from the T5-small model. On the other hand, we also showed that adopting an Encoder-Decoder architecture with BERT is not an optimal solution from the moment that BERT-base requires large amount of data and longer trainings.

Once we had complete our QA module, main focus of our exam project, we started working on the Project Work extension direct at fine-tuning the BERT-Multilingual model on the translation task. Unfortunately, our experiment exhibited sub-optimal performances when trained with a limited dataset. This observation underscores the necessity for a more extensive range of examples during the training process. The model's sensitivity to dataset size indicates that it thrives when exposed to a larger and more diverse dataset.

Our hypothesis is that the structural disparities between the Chinese and English languages contributed to the weak performances in translation tasks involving these two languages. The model's inherent capacity to capture linguistic nuances across languages might not be adequately addressed in cases with significant structural differences, cause the model not only has to learn to translate, but also how to represent those words and/or sentences in a different alphabet and with

completely different grammatical structure.
In the end our fine-tuning efforts may have been insufficient to fully optimize BERT-Multilingual for this specific translation task. While the potential of fine-tuning is evident, striking the right balance and parameters for an effective outcome is an ongoing process that warrants further investigation with tools that were not available to us.
On the positive side, aware of these consideration from our failed experiment, we decided to adapt the T5-small model for the translation task as well. This time we decided to focus on the Italian-English pattern in the hope that a more similar sentence structure would allow better results respect to ones observed with BERT-Multilingual. Our expectations have been only partially fulfilled: the model fine-tuned to perform the translation from Italian to English performed well, with promising results and good scores; but the same model re-trained on the inverse translation task performed poorly. This strange behaviour might be explained with the fact that the starting T5-small model has been previously trained on non-Italian data; thus the encoder part of the model performed well only when had to generate English text.

## 7 Conclusion

Our experiments showed the possibility of implementing a functional chatbot by, from one side, fine-tuning LLMs to complete some task such translation and summarization, and from the other side to combine them with other techniques, such Semantic Search, to reduce the computational costs required by the overall system.
We also made use of Pincecone to store our dataset of embedded Wikipedia documents, making it available for a different range of applications without the need of storing the dataset on a specific device.
After conducting all the experiment described in Sec.5, we created a prototype version of the chatbot utilizing the best performing models obtained in our study.
In conclusion, we showed the possibility to realize a functional chatbot aimed to help tourists during their travels in a foreign city, that can also be easily adapted to a wide-range of topics by changing the document collection that provides the relevant information for the specific topic of interest.

## 8 Links to external resources

- The complete project code can be found on GitHub. The notebooks used for the described experiments can be found in the relative folder, while the file *BoloBot-prototype* is a brief demo of the whole system working.
- Part of the code implemented in the the project has been adapted from the examples in the Pinecone tutorials and from "An Introduction to Open Domain Question-Answering".
- The experiments have been developed starting from some tutorials available on Huggin Face: summarization, translation and encoder-decoder models.

## References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer.