**CREATE CHATBOT IN PYTHON**

**Team Member: aut104302 SIVAVETRIVEL M**

**Project title: CHATBOT IN PYTHON**

**Phase 3: Development Part-1**

**Topic: Start measure energy consumption by loading and pre-processing the dataset **

**Introduction:**

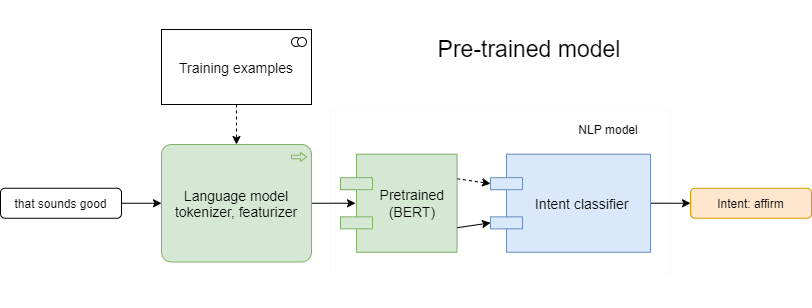
To get a well-performing chatbot with accurate [intent classification and question-answering](https://intersog.com/blog/the-basics-of-qa-systems-from-a-single-function-to-a-pre-trained-nlp-model-using-python/), you usually need a massive amount of training data. Generally speaking, that data is collected from human inputs (dialogues and chats with the bot) and used to retrain the underlying model.

At the beginning of the [chatbot development](https://intersog.com/chatbot-development-services/) process, however, you may face the lack of training data, which results in low accuracy in intent classification. A few workarounds exist to solve this problem:

1. [Pre-trained model](https://intersog.com/blog/three-methods-of-pre-processing-data-in-chatbot-development/#pre-trained-model)
2. [Training data generator](https://intersog.com/blog/three-methods-of-pre-processing-data-in-chatbot-development/#training-data-generator)
3. [Crowdsource](https://intersog.com/blog/three-methods-of-pre-processing-data-in-chatbot-development/#crowdsource)

These three methods can greatly improve the NLU (Natural Language Understanding) classification training process in your chatbot development project and aid the preprocessing in text mining. Below we demonstrate how they can increase intent detection accuracy.

**1. Pre-trained model**



The pre-trained language model can be used for NLU tasks without any task-specific change to the model architecture. Pre-trained models have an ability to continue pre-training on custom data, strarting from some checkpoint

This process is compute-intensive and requires a massively parallel compute infrastructure. The training of such general models on task-specific training data, called Fine-tuning, is far less computationally demanding and used more often. Well-known examples of similar models are Google’s [BERT](https://github.com/google-research/bert), [XLNet](https://github.com/zihangdai/xlnet" \t "_blank) and OpenAI’s [GPT-2](https://openai.com/blog/better-language-models/).

Let's find how the pre-trained model can help with intent classification. We use the following config.

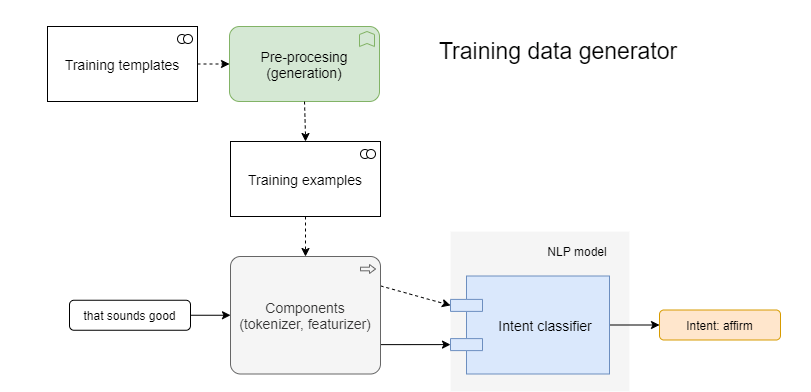
***Expected F1-score = 0.930612***

As we see, without modification of training data, usage of the pre-trained BERT model improves the accuracy of intent detection. This happens because the model already has knowledge about word's synonyms, which helped to recognize matches.

### **Fine-tuning your AI chatbot**

To perform Fine-tuning of the chatbot development model, follow the instructions on [Sentence (and sentence-pair) classification tasks](https://github.com/google-research/bert#sentence-and-sentence-pair-classification-tasks) from Google's BERT repository. In general, you need to download some text corpus or to convert your text data to BERT's input format, then run Fine-tuning command. You can prepare a new model with the following script.

## 2. Training data generator



To quickly get more training data for NLU task, you can use a training data generator. It is a program which takes templates and generates a lot of examples for model training. Well-known generators are [Chatito](https://github.com/rodrigopivi/Chatito" \t "_blank) (in [Node.js](https://intersog.com/node-js-development-services/)) and [Chatette](https://github.com/SimGus/Chatette" \t "_blank) (in [Python](https://intersog.com/python-development-services/)). A [Python programming language](https://intersog.com/blog/python-application-development-common-use-cases-and-project-examples/) preprocessor is a common tool.

Instead of writing text preprocessing examples directly, you write one or several template files in a specific format, then run the generator which parses templates and outputs ready-to-use examples. This is one of the essential preprocessing steps in text mining. To train a powerful model, you need to use the generator ot just for intent detection, but more for named entity recognition.

Let's generate samples for our easy intent with the NLP text preprocessing Python Chatette. Write a template file "affirm.chatette" with the following content:

## 3. Crowdsource in AI chatbot development

To get more training data of high quality, you can outsource jobs to distributed workers, using [Amazon Mechanical Turk](https://colab.research.google.com/drive/1KYQJj21ydDawozMQR2KyM575k1Wjqeub), [Microworkers](https://www.microworkers.com/), [Clickworker](https://www.clickworker.com/" \t "_blank) platforms.

MTurk is one of the text preprocessing tools to take simple and repetitive tasks that need to be handled manually. In a short time and at a reduced cost, you can get human-written training data for your initial model to augment training data collection and ultimately accelerate [Python-based](https://intersog.com/blog/some-cool-things-you-can-do-with-python/) chatbot development using data preprocessing for text classification.

## Key takeways

A purple and orange robot with gears and a padlock

Description automatically generated

In this article, we have considered three ways to improve language model intent detection accuracy.

**The first one**, the Pre-trained model, is the most computation-heavy yet allows us to have fewer examples to perform training, without accuracy loss.

**The second one**, Training data generator, is the simplest method that still requires to write some templates in a specific language. It reduces the amount of text input and grows the number of available samples to train the model.

**The third one**, Crowdsource, gives the most accurate output written by humans, but typically involves an extra cost to put in place.