Documentation

Hamdi Rezgui

March 2021

Contents

1 Introduction

In this document, i will present a brief description of the code files that i used to get the results of my model

2 Brief description of the code

2.1 Building the information retrieval structures

To build the vocabulary, the inverted index,document IDs the documents length and the direct structure use the "building_Trec_Collection.sh" for Trec collection , " build_wikir_collection.sh " and for wikIRS collection "build_wikirS_collection.sh " They use two classes for Trec collections defined in "Trec_Collection_opt.py" and for wikIR collections in "wikIR_Collection_opt.py" The previous two classes for the collections use an object of class "Inverted structure" and "Direct structure" that are defined in the files with the same name. NB: To generate the bag of words structure, i defined methods to do that from a generated direct structure". You will need to write a small script where you load the direct structure and generate the bag of words structure. This can be done simultaneously in with the generation of the other structures in a more efficient way. I didn't have time to do that. All these structures are computed offline and loaded whenever we need it .

2.2 evaluating the exact term matching model on the collections without training

You have to run the .sh files that start with "evaluating" in the repository in git. There is the definition of the "Queries" class that is defined in the file with the same name and which is used in the evaluation files. The queries are processed and the generation of the structure is computed on the spot and not offline This file contains a definition of non differentiable baseline IR models for exact matching like TF, TF-IDF, DIR, BM25 and JM. It also has a definition of weighted versions of those models where they multiply by the TDV weight in the computation the relavance score for each token. The weighted versions were made as an experiment. They weren't used in the end in the code.

2.3 Training

To train the models with one dense layer, you have to use the "training_trec3.sh" for a model where you can chose the models and the collections you want to train on and "t. The definition of the models are in the file "diffrentiable_models.py" To train the models with two dense layers, you have to use "training_trec_2_layers.sh" and the models are defined in "differentiable_models_2_layers.py" The code works with the latest version of tensoflow(version 2) and can be run on gpu. For wikiR and WikiRS it is similar with corresponding files.

2.4 Contact info

For any questions, you can send me an email at the following address : hamdi.rezgui1993@gmail.com