NLP/ML Assignment

Cambridge Quantum

July 29, 2021

This assignment is designed to test basic skills of the candidate with regard to NLP and ML, as well as their ability to write high-quality code and to communicate clearly research outcomes.

1 Summary

Given a simple context-free grammar, you have to to create a dataset of short sentences falling into two categories (IT and food). The goal of the project is to write, train and test a simple ML model that, given two sentences, it can detect whether they belong to the same category or not. You will use a simple compositional model for computing the internal representation of a sentence.

2 Description

2.1 Dataset

Consider the simple context-free grammar below:

S $\rightarrow$ NP VP
NP $\rightarrow$ N
NP $\rightarrow$ ADJ N
VP $\rightarrow$ VB NP

Based on that grammar and a vocabulary of your choice, write code that automatically generates a dataset with entries of the form:

sentence_1 sentence_2 label

where sentences are related to food/cooking or to IT; for example:

skillful cook prepares meal
programmer writes complicated code

Further, *label* is 1 if the two sentences are of the same category (they both talk about food or both talk about IT) and 0 otherwise.

Important: Try to keep your vocabulary short, so words are shared between different sentences and the model gets a chance to train them properly. Further, be sure that both categories share a common subset of the vocabulary, so the task is not completely trivial for the model.

2.2 Model

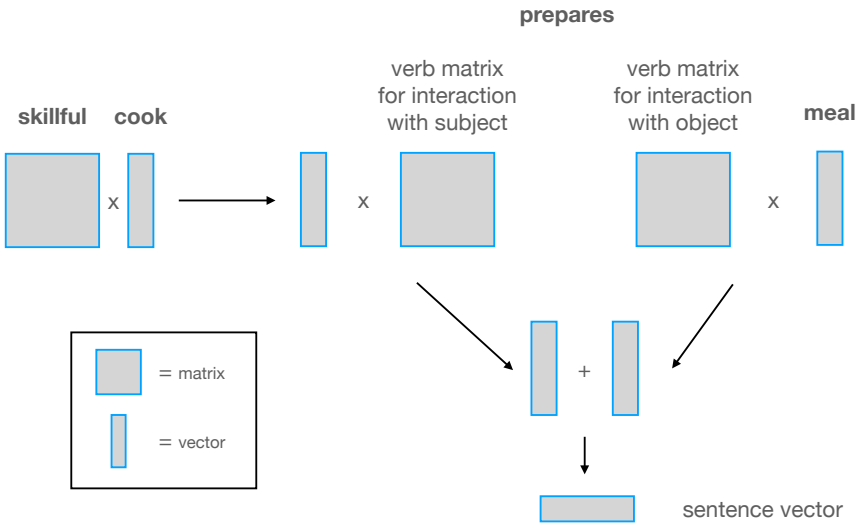
Based on the above dataset, write and train a supervised model that, given two sentences, it detects whether they belong to the same category or not. The model makes the decision by preparing and comparing compact states of the two sentences, using tensor representations of the words. Specifically:

- Nouns are represented as vectors in $\mathbb{R}^d$;
- Adjectives are represented as matrices in $\mathbb{R}^{d \times d}$;
- Verbs are modelled by two matrices in $\mathbb{R}^{d \times d}$, one responsible for the interaction with the subject, and one responsible for the interaction with object.

The vector $\mathbf{s}$ of a sentence is computed as:

$$\mathbf{s} = \mathbf{V}_{sbj}\mathbf{sbj} + \mathbf{V}_{obj}\mathbf{obj} \tag{1}$$

where $\mathbf{V}_{sbj}$, $\mathbf{V}_{obj}$ are the matrices for the verb, and **sbj**, **obj** the vectors for subject and object (possibly generated by a prior interaction of an adjective with a noun). A composition example is given below for the sentence "skillful cook prepares meal":[1]



After computing vectors for the sentences of a given pair, you should compare them and decide for a class label using an appropriate method and objective function of your choice.

2.3 Comments

- The code must be written in Python 3.x. Please **do not** use notebook format.
- Try to write your code to a high standard, as it would be intended for using it in a professional environment (as opposed to an academic environment).

[1] While the verb/subject interaction in the above diagram is depicted transposed (i.e. as $\mathbf{sbj}^{\mathrm{T}} \cdot \mathbf{V}_{sbj}$), this is only in order to keep the diagram simple while respecting the order of the words. The subject component should be typically computed as $\mathbf{V}_{sbj} \cdot \mathbf{sbj}$, as in Equation 1.

- Split your dataset in train, dev and test parts according to standard practice. Be sure the size of each part is sufficient for the task at hand.
- You can perform the optimisation by using a library of your choice (e.g. PyTorch), or write your own code if you prefer.
- Choice of hyper-parameters, dimensionality of tensors, even loss function is up to you.

2.4 Deliverables

Please send us the following:

- A zip file containing all the code you wrote for this assignment, including the module for creating the dataset and the testing code.
- A zip file containing all the datasets in text format.
- A report in PDF format with the following parts:
  - the total time you spent in this assignment
  - a short discussion explaining any design choices in the model, e.g. choice of loss function, use of regularisation or not and so on
  - a short discussion on the results
  - optimisation plots showing the convergence of the model
  - a brief section explaining an alternative approach to the problem of your choice and comparing it with the proposed method.