NLP Project: Quora duplicate detection

Justin AYIVI, Samy FERRAT, Othmane ZARHALI and Sebastien CHAILLOU
Université Paris-Dauphine

I. Dataset presentation

The dataset is a question set from the Quora site, it is broken down into a train set of 404290 rows and a test set of 1048574 rows.
It is composed of 4 columns 'questionsid': Identifier of the pairs of columns, (qid1,qid2): Corresponding to the individual identifiers of the questions (only for the ream), ('question1','question2'): text corresponding to each question, 'is duplicate': Target variable that indicates whether the two questions have the same meaning.

II. Preprocessing

A. Text cleaning

To perform the dataset preprocessing, we first proceeded by "cleaning up" the text, i.e. we replaced contractions with their original forms using a predefined dictionary (for example "can't" with "cannot"), we also replaced common errors or different forms of writing the same thing (eg bestfriend with best friend), this then helps to avoid unnecessary confusion for the model.

B. Word2Vec

Concerning the Word Embedding, we used the Skip-Gram variant of the Word2Vec algorithm, the Word2Vec algorithm does not require any labeling, it is an unsupervised learning algorithm, the Skip-Gram variant seeks to predict the words of the context from a central word.

Indeed, the outputed train set and validation set are 2D tuples compound of:

- (Question1,Question2) vectors
- is_duplicate boolean

Here is a synthesized vision of the preprocessing step:

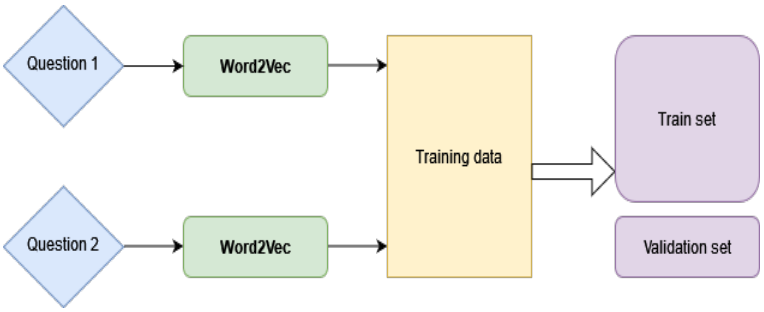

Fig. 1: Preprocessing step

III. Evaluated models

In the literature, many models have been tested with and without a RNN based (LSTM most of the time) attention models. For instance, in [1] we have the following performances:

Model Class	Model	Test Results Accuracy (%)	F-score
	Most frequent class	63.1	-
Linear	LR with Unigrams	75.4	63.8
	LR with Bigrams	79.5	70.6
	LR with Trigrams	80.8	71.8
	LR with Trigrams, tuned	80.1	71.5
	SVM with Unigrams	75.9	63.7
	SVM with Bigrams	79.9	70.5
	SVM with Trigrams	80.9	72.1
Tree-Based	Decision Tree	73.2	65.5
	Random Forest	75.7	66.9
	Gradient Boosting	75.0	66.5
Neural Network	CBOW	**83.4**	**77.8**
	LSTM	81.4	75.4
	LSTM + Attention	81.8	75.5
	BiLSTM	82.1	76.2
	BiLSTM + Attention	82.3	76.4

Fig. 2: Models evaluated in the [1]

A a result, the models that seem to perform well are those based on three main complexity levels :

- **Level 1**: feature encoding via a word embedding technique
- **Level 2**: Memory dependency, which is obvious for a language model
- **Level 3**: Context inclusion via attention models

In this project, we have performed a tailored language model that incorporates the level 1 and 2:

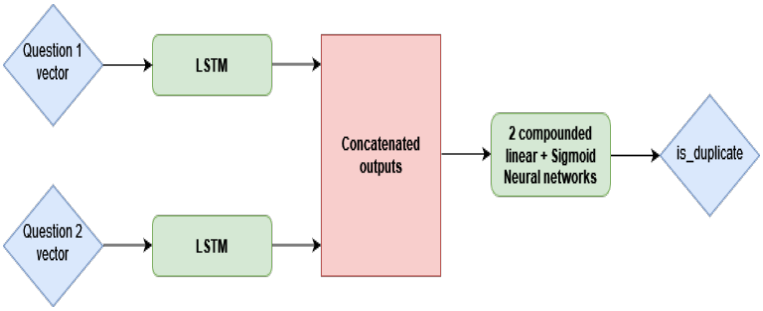

Fig. 3: Language model

IV. Code architecture

The code is decomposed into different entities summarized as follows:

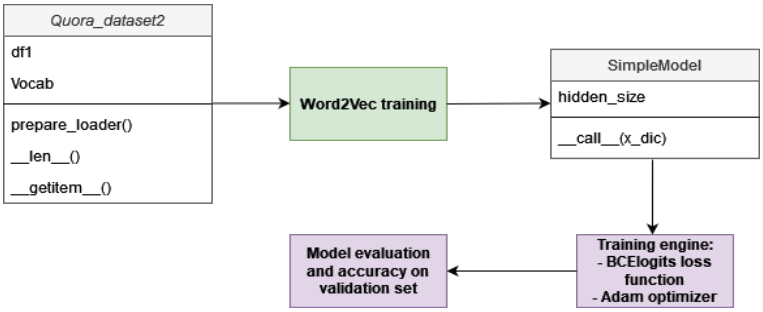

Fig. 4: Language model

V. Results

Using the previous language model, we reached an accuracy on validation set of 72.10%. This accuracy can be enhanced taking into account the following tips:

- Training on a larger dataset
- incorporating an attention model
- Training using an elaborated optimization algorithm
- Adding complexity in the neural networks (depth and highly non linear activation functions)

***p.s**: the code available here: https://github.com/SebastienChaillou/Quora_questions_NLP*

References

[1] Lakshay, S., Graesser, L., Nangia, N., and Evci, U. *Natural Language Understanding with the Quora Question Pairs Dataset*