Artificial & Computational Intelligence / Published Online : June 2020
https://acors.org/ijacoi/VOL1_ISSUE3_14.pdf

Artificial & Computational Intelligence

Question Duplication using Deep Learning

Anudeep Atyam[1], T.Suvarna Kumari[2], Hephzibah Saidu[3]
Department of Computer Science
atyam.anudeep@gmail.com[1], suvarnakumarit@gmail.com[2], hephzibahsaidu98@gmail.com[3]
Chaitanya Bharathi Institute of Technology, Hyderabad, India

Abstract

Questionnaire platforms like Quora, Stack-overflow who are based on functionality of allowing users to ask questions and allow them to respond to questions revolve around a frequent and well raised problem of semantic matching or question duplication. We often tend to write a sentence or two which are definitely influenced by the linguistic nature we live around to the accent we use to the environment we live in. These habits might tend to produce a major drawback for users to ask questions which mean the same but are not identified as duplicate questions. We have come up with a pattern which may not completely provide a solution to this problem but may help in increasing the efficiency of the model to predict the duplicate ness among several question pairs. We propose to use a Deep Learning approach LSTM, long short term memory as the base model in a layer of subsequent sequential model with a combination of five layers like dropout layer, embedding layer, dense layer. We just tried to prove that a machine learning model like Random forest did not perform well when compared with a Deep Learning model LSTM mainly because of non consideration of order of words. We tend to propose some basic data pre-processing techniques which pave the way to increase the efficiency of the input data thereby increasing the efficiency of the model in identifying the delicateness of the question pairs in the form of accuracy.

Keywords: Long Short Term Memory, RNN, Question Matching, Natural Language Processing.

1. Introduction

For a well sustainable platform, maintaining the data accurately is the fundamental task. With increasing loads of data every day Questionnaire platforms seem to stumble upon main concern of duplicativeness in Questions or the answers present .This may lead to scattered information into different branches leaving the user to access only one of the nearest question/answer and making them completely unaware of the answers divided across duplicate questions leading to insufficient outcome. The whole agenda of these questionnaire platforms is to make the data available for every question which is lost by this major problem of Question Duplication. For example, we'd consider questions like “strategy of 2015 elections”, “trumps strategy of winning 2015 elections”, and “how did trump win 2015 elections” are the questions which intent the same but are considered as different questions. To solve this problem of question duplication , we've proposed a Deep learning algorithm to automatically identify the probability of duplicativeness among the question pairs.

We have considered several reasons which may lead to Question Duplication problem , one such problem which has constituted the majority of effect in the question duplication problem is considering ordering of words placed in a question, which is not considered by machine learning algorithms like Random Forest etc . For example:"Man bites Dog","Dog bites Man". These questions when observed have the same words which may be considered as duplicated questions when order of words

in particular is not taken into account leading to an inefficient outcome.

2. Related Work

As the problem of Question duplication is spread across various applications , the study on finding an accurate solution has been for a extended time.[1]Previous work to identify duplicativeness among question pairs based on the semantics of the question pairs followed some traditional machine learning algorithms like Support Vector Machines(SVMs).[2]But from the emergence of Deep Learning , Artificial Intelligence, Neural networks and Natural Language Processing techniques, these wide range of models have shown some impressive results. Convolution Neural Networks(CNN) [3]have shown good results in few sentimental analysis tasks where we have identify the true nature of the sentences by weighing them against the semantics used in the phrase, and classification tasks. However, most of the ar Xiv:1801.07288v3 [cs.CL] 28 Jan 2018 [4].Deep learning methods proposed for validating duplicativeness among the sentences have used a Siamese Neural Net-work architecture makes use of a neural network for extracting features from the input sentences and then compares with the help of a distance metric[5].Neural networks played a prominent rule in wide range of NLP tasks. A Siamese neural network which will split into two sub neural networks for processing and the outputs from both of them will be combined was proposed[6].Although because of its lightweight and straightforwardness to train the model, there is no specific interrelation which allows information not to lose.Inorder, to solve the drawbacks of the Siamese neural network, a Compare-Aggregate model was proposed , which observes and note the similarity between two sentences. Data science engineers at Quora recently released a public dataset of duplicate questions that's wont to train duplicate question detection models [7]. Research at the Department of Computer Science, Stanford University [8] and ny University was an excellent source of knowledge for us to dive into. Evaluation of a model for extracting various features [9] was administered which incorporates the concept of fuzzy and vector distances of the texts also [10].

3. Existing System

Existing systems focus more on the future input data eliminating the focus on already present data.

The algorithms which are in use are developed based on machine learning models. Which might produce outputs which are not consistent? The data pre-processing techniques used in the data selected is focused on the grammar and root words of question pairs.

- Ordering of word is not considered
- Cannot be applied on already present data
- Consider very limited features and usability.

4. Proposed System

We worked on a model based on LSTM , long short term memory algorithm to provide a solution for question duplication problem faced by many Questionnaire platforms. We developed a model which is a stack of sequential layers like embedding layer, dropout layer, lstm and dense layer. It concentrates on order of words in the input and learns a high level presentation of everything in all layers and makes use of those features to calculate the final probability of duplicativeness.

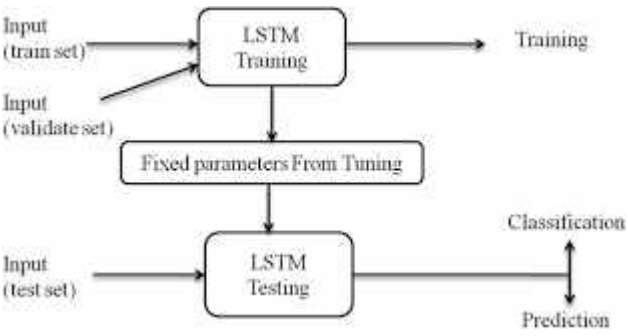

Fig. 1. Block diagram of proposed system

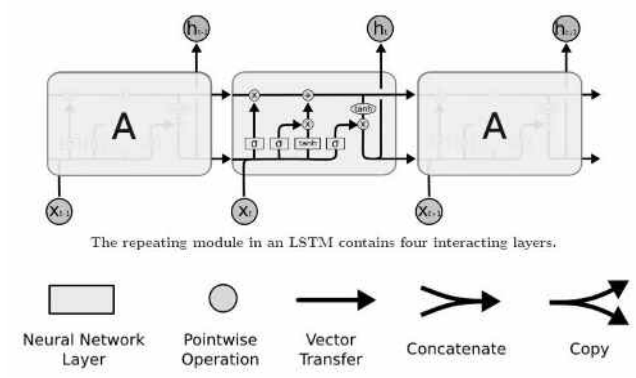

Fig. 2. Architecture of LSTM

A. Components

Data Extraction: For training and testing of our model we have used two different data sets which were given by Quora. It contains a total of 404,290 question pairs. It contained the following columns labeled as : “id”, “qid1”, “qid2”,“question1”, “question2” and “is_duplicate”. Testing dataset does not have is_duplicate column which is the output , probability of duplicativeness of the question pairs. We have only considered questions and a unique id for a question pair as features.

Data Pre-processing: Pre-processing makes or breaks the output of the model as it is the most crucial step of any predictive analysis problems. Understanding the data by doing few visualizations and behaviour of one column with another. Here our dataset did not contain much of the important features and it was completely a straight forward dataset. Hence we directly chose our features. We identified a few null values and dropped them by using a lambda function. We developed a function to fill acronyms and to replace few words like "100" as hundred , "usa" as united states and so on.

Feature Extraction: After we perform data preprocessing techniques in order to identify the exact features from the data. The data we used from Quora dataset contain direct features which didn't require any feature extraction. We chose the question pairs as features of inpt and the column which predicts whether the question pair is duplicate or not as the feature of output vector. This process has a direct impact on the accuracy and efficiency of the model prediction.

Python pickle: Pickel is a module used for object serialization. It converts objects into binary format and from binary to objects.Pickel , an object structure have two process to name after,.“Pickling” ,the process whereby a Python object is converted into a byte stream, and “unpickling” is that the inverse operation, whereby a byte stream (from a computer file or bytes-like object) is converted back to an object. Pickel is a process which can also be referred to as "serialization","marshalling".

hStack: Arrays are stacked sequentially during a column wise manner. This is much like concatenating the second axis , apart from a one dimensional array which has the first axis in situ of second. It also rebuilds or regenerates hsplit.

Snowball Stemmer: The algorithm of snowball stemmer is developed by Martin Porter.

It supports a huge number of languages and does not narrowed down to just English language which allows a huge weightage over regular porter stemmer. The name stemmer is derived from the developers name Porter as he created a programming language with all new stemming algorithms.

Main feature of this Snowball stemmer is that Porter himself stated that Snowball Stemmer is far more efficient than Porter Stemmer which only considers the prefix and suffixes of the words to their root word.

Padding sequences: Padding in a brief sense means adding extra bits to an original value to normalize it to a specific range. In sentences there might be a wider possibility than few sentences might be short and few might be very humongous . the length of the sentences does not depend on any particular feature. Hence comparing or performing operations, processing these data might in uneven for the model. Hence padding all the sentences or input data into a normalised vector value which can be used as a constant rate for the model and hence can work very efficiently in case of any input. Padding can be done according to our desired length, if less value is used as padded value then the extra data can be truncated and vice versa.

Texts_to_sequence: Classification of sequences is a key role in predictive modeling problems. These problems mostly revolve around input which have to be united into a specific sequence in order to perform the necessary operations and conclude on one of the outputs. Regularly used to convert a scattered set of data into single sequence so that both of the inputs and outputs are in a singular axis or type and can be compared or processed. The sequence can be formed using NLP modules or any other modules too.

Word Embedding: Word embedding is a technique where we map each value in the sentence whether it may be a root value or an alternatively used word to that of a real world vector . In other words word embedding is normalization of all the words present in the dataset so as

to maintain a range which completely neutralizes the scope for the model and increases the efficiency additionally. The vector values are the high dimensional vector values, the word can map to.

Stopwords: Terms like “stop words” ,“stop word list” ,“stop list” refer to same thing and is commonly referred as stop words. The term direct to group of words in any language not only limited to English. Stop words play a very crucial role in certain applications because they add on weight to a sentence by hiding the important words and weighing them down. Because of their wide usage it is applicable to a whole variety of applications. Consider an example , if we perform a search operation of" how to develop a search engine", if the model focuses on "how","to", "a" more than "develope","search","engine", it will automatically be over weighed and will tend to give results which may not be related to the search operation . Hence identifying and removal of stop words add up to a highly efficient solution in terms of a semantic based model.

5. Methodology

The proposed system consists of all the above mentioned components in the IV Section. The data is passed through each and every component respectively to maintain the structure of the data before it is trained by the Deep learning model . The data fed into the model can affect the efficiency if it is not passed through the above components. A deep Learning model which is sequential id built with five layers to start with. We have used 5 different layers

Which includes drop out layers, lstm as on layer , an embedding layer and a dense layer.

Clear process of all the steps taken in the process of training and testing the model is clearly depicted in the following fig(fig.3) . This entire process completely affects the outcome of the result.

The layer wise depiction of the sequential model has been explained in the following image (fig.4)

The flow of data from the initial phase to result is as follows:

1. With the data obtained from the Quora dataset, we have worked with 2 datasets training dataset and testing dataset.

2.We used only three features from the entire dataset which would be 2 question pair columns and a output column.

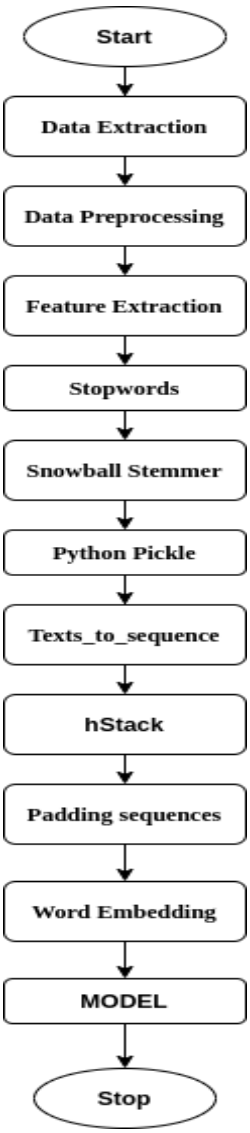

Fig. 3.Flow chart of working mechanism

3. The data is now cleansed by removing null parameters, replacing few acronyms, substituting values for numbers. Regular expression of the sentences is also taken into consideration.
4. This entire cleansing process is now carried now for both the testing dataset and training datasets.
5. The data is now passed through snowball stemmer to eliminate the non root words by replacing them with root words
6. This data is now passed through the stop words module which eliminates all the stop words which weight the sentence unnecessarily.
7. The cleansed data is now passed through pickle, a serialization function which converts objects to binary and wise versa.

8. We opted for hstack which would sequentially arrange the data in column wise manner and this data is passed through text_to_sequences.performed for both test data and train data.
9. The max length of the word in the entire dataset is obtained by using a lambda function which is given both testing an training datasets
10. The max length is used for padding all the words in the dataset.
11. This padded data is passed through model which is elaborated in below fig 4

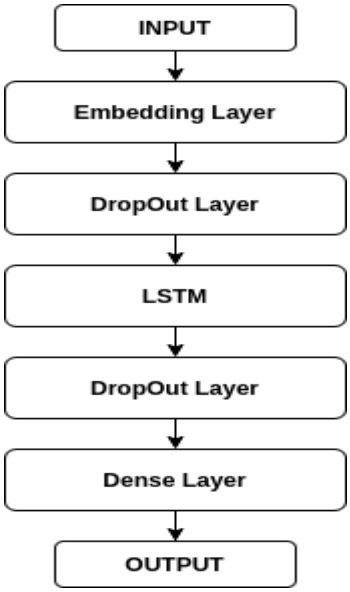

Fig. 4.Flow chart of model developed

6. Results

The proposed system has given some efficient results. We have stored the probabilities of duplicativeness among the question pairs into an output file which is shown fig 5(see below).

The summary of the entire model we generated with their briefing is displayed in the fig 6(see below).

To start with the embedding layer which has been used for optimization of the data into a certain range which affected the accuracy of the model prominently?

Dropout layers are used to eliminate over fitting which is considered as the main effect for the huge amounts of data present in questionnaire platforms

Third layer consists of LSTM, which will do the actual processing of considering the order of the words in the given question pairs to find the duplicativeness.

We have used another dropout layer which has shown much more effective results which have taken place during testing.

We rounded up by using a dense layer which used rmsprop for normalization of the data.

	A	B
1	test_id	is_duplicate
2	0	0.044398803688852
3	1	0.525038303266034
4	2	0.595735477884185
5	3	0
6	4	0.556195914858334
7	5	0.010026526741041
8	6	0.625830668961597
9	7	0.325833516267358
10	8	0.567440492446129
11	9	0.004284138051501
12	10	0.640683007629854
13	11	0.003931641029928
14	12	0
15	13	0.564736348749649
16	14	0.349274311167315
17	15	0.598052738628397
18	16	0
19	17	0.616877919511787
20	18	0.662579305850007
21	19	0.619709251448825

Fig.5 Output Data

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, None, 32)          16000
_________________________________________________________________
dropout_1 (Dropout)          (None, None, 32)          0
_________________________________________________________________
lstm_1 (LSTM)                (None, 32)                8320
_________________________________________________________________
dropout_2 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 33
=================================================================
Total params: 24,353
Trainable params: 24,353
Non-trainable params: 0
_________________________________________________________________
```

Fig. 6 Model summary

The results of each and every layer in percentages have

been displayed in the fig 7(see below).We have performed five epochs which have given an accuracy of around 78%. If the epochs are increased along with layering of the model then the model would definitely give around an acceptable accuracy to be able to use in real time problems.

```
Epoch 1/5
404290/404290 [==============================] - 1194s 3ms/step - loss: 0.5339 - accuracy:
0.7349
Epoch 2/5
404290/404290 [==============================] - 1219s 3ms/step - loss: 0.5061 - accuracy:
0.7551
Epoch 3/5
404290/404290 [==============================] - 1209s 3ms/step - loss: 0.4908 - accuracy:
0.7652
Epoch 4/5
404290/404290 [==============================] - 1177s 3ms/step - loss: 0.4759 - accuracy:
0.7734
Epoch 5/5
404290/404290 [==============================] - 1204s 3ms/step - loss: 0.4618 - accuracy:
0.7811

<keras.callbacks.callbacks.History at 0x7fa7b47f2780>
```

Fig.7 Results per layer in the model developed

7. Conclusion and Future Work

The usage of a Deep Learning technique LSTM have given a significant amount of accuracy with just 5 layer models and within 5 epochs. It also gives reliable outputs when the ordering of words may cause the confusion of duplicativeness to the model.

Using LSTM as the main model we are concluding that LSTM gives a higher level of reliable results than the referred models when solving the problem of Question Matching.

We have compared between a machine learning algorithm Random Forest and Deep Learning Algorithm, LSTM to obtain a clearer picture of the solution. Shown in Table.1

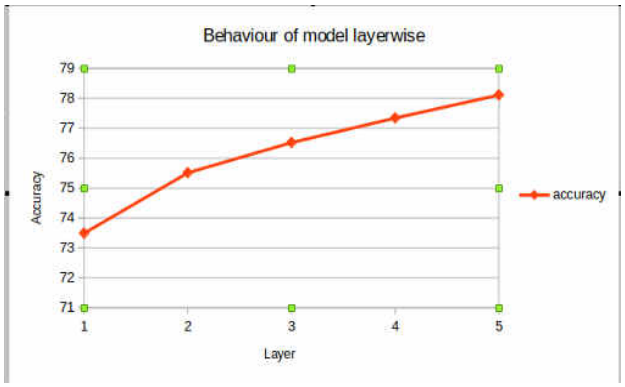

Fig.8 Graphical representation of Accuracy behaviour Layer-wise of model

TABLE.1 Comparison between Machine Learning Model and Deep Learning Model On Duplication Problem Considering Order Of Words

Model used	Results (percentage)
Random Forest	71
Lstm	78.11

References

[1] Mart´ın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado,Andy Davis, Jeffrey Dean, Matthieu Devin, et al.2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprintarXiv:1603.04467

[2] YEUNG, K. (2016, March 17). Quora now has 100 million monthly visitors, up from 80 million in January. Retrieved from venturebeat.com: https://venturebeat.com/2016/03/17/quora-now-has-100-million-monthly-visitors-up-from-80-million-in-january

[3] Lili Jiang, S. C. (n.d.). Quora. Retrieved from engineering.quora: https://engineering.quora.com/Semantic-Question-Matching-with-Deep-Learning

[4] mccormickml. (2016, April 12). Retrieved from mccormickml:http://mccormickml.com/2016/04/12/googles-pretrained-word2vec-model-in-python

[5] Machine Learning Mastery. (2017, June 15). Retrieved from Machine Learning Mastery:https://machinelearningmastery.com/prepare-text-data-machine-learning-scikit-learn.

[6] Brownlee, J. (2017, October 9). A Gentle Introduction to theBag-of-Words Model. Retrieved from machinelearningmastery.com:https://machinelearningmastery.com/gentle-introduction-bag-words-model

[7] Rajaraman, A.; Ullman, J.D. (2011). "Data Mining". Mining of Massive Datasets (PDF). pp. 1–17.doi:10.1017/CBO9781139058452.002. ISBN 978-1-139-05845-2

[8] Gilyadov, J. (2017, March 23). Word2Vec Explained.Retrieved from github.io: https://israelg99.github.io/2017-03-23-Word2Vec-Explained.

[9] McCormick, C. (2016, April 12). Google's trained Word2Vec model in Python. Retrieved from mccormickml.com: http://mccormickml.com/2016/04/12/googles-pretrained-word2vec-model-in-python

[10] Thakur, A. (2017, Feb 27). Is That a Duplicate Quora Question? Retrieved from Linkedin: https://www.linkedin.com/pulse/duplicate-quora-question-abhishek-thakur

[11] Github. (2017, April 16). Retrieved from github: https://markroxor.github.io/gensim/static/notebooks/WMD_tutorial.html

[12] E. Agirre, C. Banea, D. Cer, M. Diab, A. Gonzalez Aguirre, R.Mihalcea, G. Rigau, and J. Wiebe, "Semeval2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation," in Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), 2016, pp.497–511

[13] E. Agirre, A. Gonzalez-Agirre, D. Cer, and M. Diab, "Semeval-2012 task 6: A pilot on semantic textual similarity," in Proceedings of 1st Joint Conference on Lexical and Computational Semantics, 2012, pp. 385–393.

[14] Andrei Z Broder. 1997. On the resemblance and containment of documents. In Compression and complexity of Sequences 1997. Proceedings. IEEE, pages 21–29.

[15] Kuntal Dey, Ritvik Shrivastava, and Saroj Kaushik. 2016. A paraphrase and semantic similarity detection system for user generated short-text content on microblogs. In COLING. Pages 2880–2890.

[16] Tim Rocktaschel, Edward Grefenstette, Karl Moritz Hermann,Tomas Kocisky, Phil Blunsom. Reasoning about entailment with neural attention. In ICLR 2016

[17] Bromley, Jane, et al. "Signature Verification Using A "Siamese" Time Delay Neural Network." IJPRAI 7.4 (1993):669-688.

[18] Wang, Zhiguo, Wael Hamza, and Radu Florian. "Bilateral Multi-Perspective Matching for Natural Language Sentences." arXiv preprint arXiv:1702.03814 (2017).

[19] Wang, Shuohang, and Jing Jiang. "A Compare-Aggregate Model for Matching Text Sequences." arXiv preprintarXiv:1611.01747 (2016).

[20] Lei Guo, C. L. (2017, Jan 16). Duplicate Quora Questions Detection. Retrieved from semanticscholar.org: https://pdfs.semanticscholar.org/4c19/2b8f45b1e913ee7da32624cd7559eccb0890.pdf