QUORA INSINCERE QUESTIONS CLASSIFICATION

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE

OF

MASTER OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by:

MONA SINGH

2K17/CSE/10

Under the supervision of

MANOJ KUMAR

(Head, Computer Centre and Associate Professor (CSE))

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

JULY, 2019

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, Mona Singh, Roll No. 2K17/CSE/10 of M.Tech (Computer Science and Engineering), hereby declare that the project Dissertation titled "**Quora Insincere Questions Classification**" which is submitted by me to the Department of Computer Science and Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirements for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any degree, Diploma Associateship, Fellowship or this similar title or recognition.

Place: DTU, Delhi Mona Singh

Date:14-07-2019 (2K17/CSE/10)

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CANDIDATE'S CERTIFICATE

I hereby certify that the Project Dissertation title "**Quora Insincere Questions Classification**" which is submitted by Mona Singh, Roll No 2K17/CSE/10 Computer Science and Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirements for the award of the degree of Master of Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi

Date:14-07-2019

MANOJ KUMAR

Head,

Computer Center and Associate Professor

Department of Computer Engineering

Delhi Technological University

ABSTRACT

The competence of any online website surely depends on the kind of experience it gives to its users which depends by and large on the content they put up on their website. Hence, the content which is being put online should be really taken care of. There are many websites which provide content to their user in terms of questions and answers for example the online website Quora, which has large scale of data in terms of questions and answers of users. Users are the ones who put up questions and also provide answers to those questions. Most people on Quora are well-meaning and are genuinely interested in asking questions. Less often, in a way that is deliberately provocative, somebody will ask a question where the wording is intended to make its own statement. This may include framing or calls for hateful stereotypes to be confirmed. These questions are harmful to our community, and we remove or hide them whenever we become aware of them. In this paper a system is proposed that will take significant amount of data from quora and use that data for different approaches to predict if the question is insincere. This project aims to develop models that take the text of a question as an input in English and produce a o or 1 that corresponds to whether the question should be approved as “sincere” or flagged as “insincere”.

ACKNOWLEDGEMENT

I am most thankful to my family for constantly encouraging me and giving me time and unconditional support while pursuing this research.

I wish to express my deep sense of gratitude and indebtedness to Mr. Manoj Kumar, Associate Professor, Department of Computer Science and Engineering; for introducing the present topic and for his inspiring guidance, constructive and valuable suggestions throughout this work. I am heartily grateful for his guidance and support during my project. His able knowledge and expert supervision with unswerving patience fathered my work at every stage, for without his warm affection and encouragement, the fulfilment of the task would have been very difficult. I would also like to extend my heartfelt gratitude towards all the members of the Department of Computer Science for gratuitously helping me in the successful completion of the project. I am genuinely appreciative of all my friends for their suggestions and moral support during my work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 GENERAL INTRODUCTION

Quora is a website created by Adam D'Angelo and Charlie Cheever, they used to work in Facebook. It claims to have data starting in June 2010 on more than 450,000 subjects, almost all published by its registered users. Type it into the search box, whatever your question, and if there is no response already, consumers will pile in an attempt to reply. Information is structured more like Wikipedia than Google, but the site utilizes the following style of Twitter to monitor the finest contributors. Quora is a knowledge gaining and sharing platform where you can ask questions from distinct sources and get responses. One aspect that differs Quora from most other question-and-answer sites, like Yahoo's long-standing responses, or WikiAnswers, is that users are supposed to use their true names often derived from Facebook or Twitter. Quora has not yet been victimized by spammers, even though they will surely test it as quickly as it gains volume and popularity, either damaging it or making it much, much better. It also had some notable downtime as it was struggling with its sudden increase in popularity.

In some ways, most of the big websites can be viewed as information filters of one kind or another. Google uses factors such as links to a web page or the reputation of the publisher to find places to answer a query. According to social connections, Facebook organizes music, topical news and love and career bonds. Twitter users organize reader esteems or hashtags around people. Quora's filter often appears to be its own members and personal experiences.

The Internet has a nagging problem: lots of information is available, but there is often confusion about what is true. Many large websites with their services are trying to solve this problem. Any major website today is tasked with the problem of creating safe content for its users. This could be anything from reporting inappropriate videos to deleting toxic and harmful comments. Quora is one of these websites that is looking to create safe content for its 300 million monthly users [1]. On their website, users are able to post questions and receive answers from anyone in the Quora community. The key challenge for Quora is to 'weed' out 'insincere' questions that are founded on false premises, or that intend to make a statement rather than look for answers [2]. Due to the size of the data and number of questions that are posted on Quora, manually classifying questions is simply not feasible. Same goes with Quora. In order to make users safe to share their knowledge. The only intention of a question should be to get answers, not to make a statement or argue a point. Questions should not be used to promote a business or service, or to make a statement about something. We can say that if a question is trying to do anything else apart from obtaining information that is an insincere question.

As Quora grows, it becomes increasingly important to identify and remove as quickly as possible insincere and trolling questions before they can cause harm (or spawn imitators). Holding online conversations from becoming toxic is one of today's most important internet challenges. This paper is an attempt to propose a system that will try different approaches to predict if the question is insincere. This is a very relevant issue today, with online forums that act as platforms for people to resolve their curiosities— such as Reddit, Yahoo Answers, and StackExchange — combating posted questions that violate the forum's guidelines. Although there are often people who act as moderators, usually people who are passionate about the topic of discussion, the number of monitoring questions far outweighs these moderators ' capacity. The ability to automatically flag disingenuous, malicious, discriminatory issues broadly, "insincere" issues will help these communities quickly remove such negative content and deter potential posters.

1.2 QUORA BUSINESS MODEL

The main source of growth for Quora has been represented by the founding coming from several rounds of investments made by venture capital firms, overall amounting to 452 million dollars. The main peculiarity, that constitutes one of the pillars on which the platform is built on, are questions answered by real people, whereas search engines as Google simply match user queries with the most relevant content already available on the web. Alongside this, Quora is an engaged community, whose members are the Quorans, where expertise can be accessed freely and easily. An additional source of revenue is generated through advertisements.

1.3 QUORA DISTINCTIVE TRAITS

Instead of being based on algorithms that rank web pages according to their relevance with respect to a certain query, Quora starts from human creation. A growing number of businesses are now adopting Chatbots and software capable of creating contents automatically, still, people exhibit the tendency to prefer interactions with other human beings and this is what Quora does: the majority of questions is about life and career advice. Users are not only looking for answers; they also want to find someone that can sympathize with them or that has already gone through what they are experiencing. Quora aims to provide value to users and to writers as well.

1.4 INSINCERE QUESTIONS ON QUORA

A question intended not to seek helpful answers but to make a statement is termed as an insincere Question. Insincere questions are not meant to ask for useful answers on a topic of interest, they can in fact be rhetorical or statements. Given this peculiarity, people can distinguish them from usual and sincere questions by looking at some factors:

- The tone, which rather than being non-neutral is exaggerated to draw attention to a point or a position
- Their ultimate intent is to inflame, suggesting a discriminatory idea, seeking confirmation of a stereotype, insulting a person or a group, often basing upon characteristics that are not measurable
- They are generally not based on evidence, nor have a solid connection with reality
- They sometimes have sexual content to be more provocative and generate a wider response (whether disdain or shock) from the community of users

Basically, any question that is intended to anger or offend and that is not being asked for information gathering purposes.

- Why do Chinese hate Donald Trump?
- Do Americans that travel to Iran have a mental illness?

We can clearly see these questions are intended to inflame and not to gain information and need to be excluded from Quora's platform. This task is difficult because of the wide range of topics, moods, and phrases that could be considered "insincere." Some words such as "stupid" or "harmful" may often lend themselves to discriminatory comments, but descriptors that are broad such as "worthless" (for example, in the question "Why are 512 MB flash drives actually worthless these days?") or meaning-overloaded (such as in "How long does it take for milk to go bad?" or "Why is Michael Jackson's Bad so famous?") are much more difficult to extract information from.

In addition, nuances such as the tone of the question, the use of jargon, slang, or abbreviations beyond the scope of the word embedding, and the phrasing differences between statements, exclamations, and questions contribute to the difficulty in identifying a questioner's intentions.

CHAPTER 2

DATASET

2.1 OVERVIEW OF THE DATA

The Dataset used in this project has been provided by Quora for the online competition on Kaggle. Quora provided a good amount of training and test data to identify the insincere questions. Train data consists of 1.3 million rows and 3 features in it.

Data Fields:

- qid—unique question identifier
- question_text—Quora question text
- target—a question labeled "insincere" has a value of, 1 otherwise 0.

```
train_df.sample(5)
```

	qid	question_text	target
612923	78095227fa51ddfa7aa3	What is aim of ISIS?	0
669114	8309b6c28cc3071fa78e	I got a call from TCS, Wipro and Infosys? Whic...	0
181351	23711ddb43068cc6be60	How long does the international transfer take?	0
343514	434a89315d7ee2186e50	What if I falsely set my profile as 'Serving N...	0
1110267	d98fec22cddfa319d1b7	What should you consider before quitting your ...	0

Table 1: Training set example

The test data is 56.4k rows of data, it obviously does not have the target labels, .

Data Fields:

- qid—unique question identifier
- question_text—Quora question text

```
test_df.sample(5)
```

	qid	question_text
227599	9adc3b2161af7df2e67c	Which world leaders were/are more popular outs...
342521	e9682173b84bef56f7d0	Which college is more LGBT-friendly: MNNIT or ...
189693	810ff62c6b485952d744	Is it possible to have complete control over c...
308423	d213678d1ba2190c616d	Would Frozen have made less money if Kristoff ...
86436	3abb438b76a24ce4255e	How much does an MTech cost at IIIT Hyderabad?

Table 2: Test set example

2.2 EXPECTED DIFFICULTIES IN THE DATASET

2.2.1 Large Dataset

There are more than one million rows of training data. The big dataset is expected to be quite challenging. Memory errors and excessive processing times may be challenging. There are several methods to attempt to counter the big size of the dataset, including using larger information samples to train and reduce dimensionality. The selection of features and it will be essential to optimize the model.

2.2.2 Imbalanced Dataset

The dataset is extremely imbalanced, with only 6% of the target (insincere) class samples. With recall this will cause difficulties. Because of the tiny number of insincere samples, maximizing recall, or true positive rate, could be a problem here. The methods of resampling and information increase could enhance the efficiency of the model.

Imbalanced classes are a prevalent issue in the classification of machine learning where each class has a disproportionate observer percentage [3]. With only 6.6% of our target class dataset, this can certainly have an imbalanced class!

2.3 EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis relates to the critical method of original data research to identify trends, identify anomalies, test hypotheses, and use summary statistics and graphical representations to verify assumptions. It is a strategy to evaluating data sets, often using visual techniques, to illustrate their main features.

2.3.1 Here is the distribution of the target variable,

```
sns.countplot(train['target'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f861e5d4a20>
```

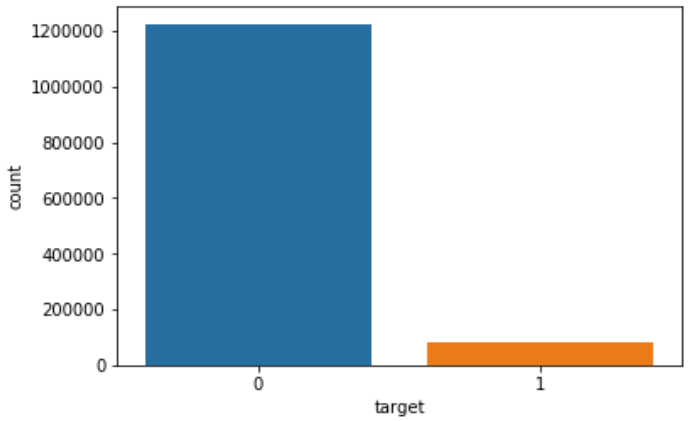


Figure 1: Graph for the distribution of the target variable

Only 6.6% of our target class dataset is 1, it is pretty imbalanced.

2.3.2 Here is the percentage of sincere and insincere questions in the given data

```
print(len(train.question_text[train['target'] == 0])/len(train['question_text']) * 100,'percent of sincere')
print(len(train.question_text[train['target'] == 1])/len(train['question_text']) * 100,'percent of insincere')
```

```
93.81298224821265 percent of sincere
6.187017751787352 percent of insincere
```

Figure 2: Output for the distribution of target variable

2.3.3 Total number of NULL values each row has,

```
train.isnull().sum()
```

```
qid             0
question_text   0
target          0
dtype: int64
```

Figure 3: Output for NULL values in each row

2.3.4 Number of tokens per question,

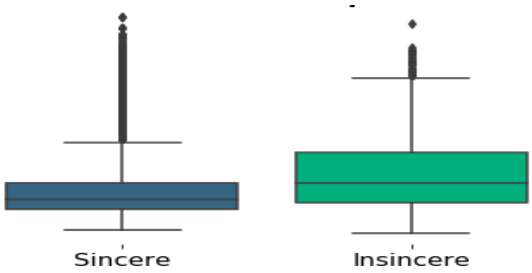


Figure 4: Graph for tokens per question

Length doesn't seem to explain insincerity, but it can be seen that insincere questions are longer than sincere questions. The plot above shows a significant difference in the number of tokens in each class.

2.3.5 Number of sentences per question,

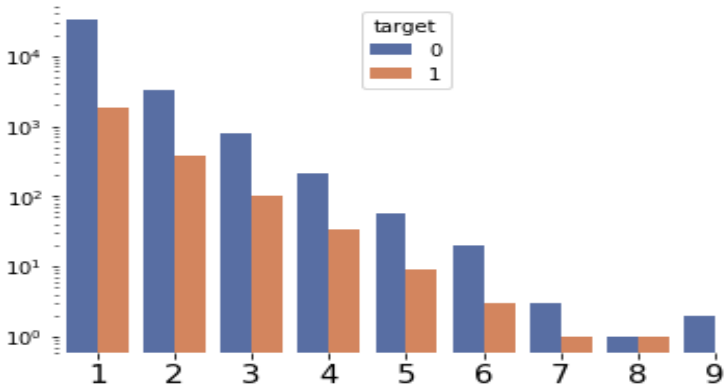


Figure 5: Graph for the number of sentences per question

Evaluating from the plot, the sentences for an insincere question is generally less than the number of sentences for a sincere question. Although there is no logical explanation for this insight in the data.

2.3.6 Insincere question common words,

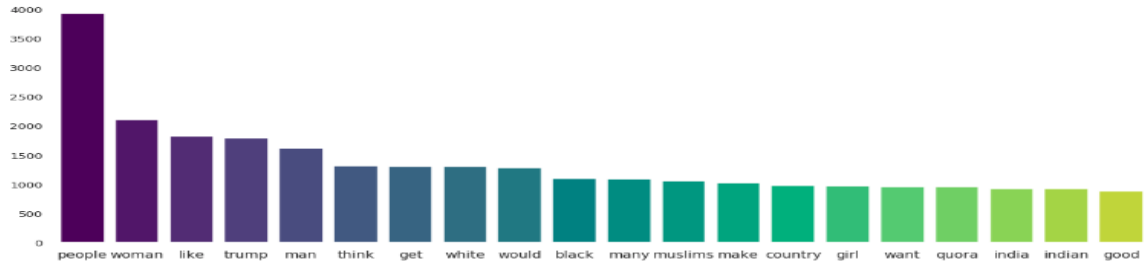


Figure 6: Graph for insincere questions common words

2.3.7 Sincere question common words,

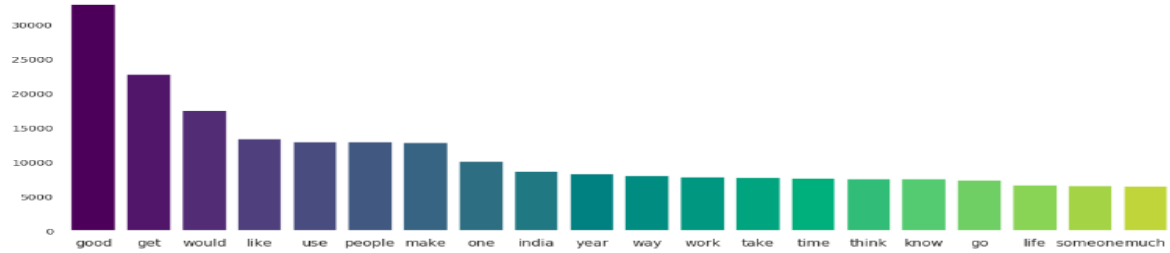


Figure 7: Graph for sincere questions common words

2.4 PREPROCESSING THE DATASET

Data preprocessing is nothing but the readying of data for experimentation-transforming raw data for further processing. For better improvement of the scores text data cleaning like data preprocessing need to done using Html tag removal, Punctuation removal (Removing unnecessary punctuation, tags), Tokenization (convert sentences to words), Lemmatization (Another approach to remove inflection by determining the part of speech and utilizing detailed database of the language.), contraction mapping. But for this database, we do not need to use standard preprocessing steps as we have pre-trained embeddings.

Pre-Trained Embeddings

The pre-trained word embedding, which uses billions of phrases, gives us the additional advantage over training our own embedding. Pre-trained models are the easiest way to get started with embedding words. A pre-trained model is a collection of words embedded elsewhere that we merely load into our computer and memory.

Pre-trained models are also available in languages other than English, opening up multi-lingual opportunities for our applications. The benefit of these models is that they can exploit enormous datasets that we may not have access to, constructed using billions of separate words, with a huge corpus of language capturing word meanings in a statistically flexible way. Examples of training data sets include the entire Wikipedia text corpus, the

common crawl dataset, or the Dataset of Google News. Using a pre-trained model removes the need for time (intensively) to acquire, clean and process such large sets of information.

Word Embeddings:

Word embeddings focus on learning distributed vector representations of words by leveraging the contextual information in large corpora using neural network architectures [4]. It is a text depiction where words with the same significance are represented in a comparable way. In other words, in a coordinate system, it reflects words in which associated words are positioned nearer together, based on a corpus of interactions.

In general, embedding represents geometric word encoding depending on how often a text corpus appears together. Various implementations of word embeddings described below differs in the way as how they are constructed [5].

Word embedding is one of document vocabulary's most common depictions. It is capable of capturing a word's background in a text, similarity between semantic and syntactic, relationship with other words, etc. They are vector representations of a particular word. The three-word embeddings are Word2Vec, GLOVE and Fasttext.

Word2Vec:

Similar words occur more often in the same context. Word2vec represents each word with a vector representation [6]. The main idea behind it is that we train a model on the context on each word, so similar words will have similar numerical representations. Just like a normal feed-forward densely connected neural network (NN) where we have a set of independent variables and a target dependent variable that we are trying to predict, we first break our sentences into words(tokenize) and create a number of pairs of words, depending on the window size. So one of the combinations could be a pair of words such as ('cat', 'purr'), where cat is the independent variable(X) and 'purr' is the target dependent variable(Y) we are aiming to predict.

We feed the 'cat' into the NN through an embedding layer initialized with random weights, and pass it through the softmax layer with ultimate aim of predicting 'purr'. The optimization method such as SGD minimize the loss function "(target word | context words)" which seeks to minimize the loss of predicting the target words given the context words. If we do this with enough epochs, the weights in the embedding layer would eventually represent the vocabulary of word vectors, which is the "coordinates" of the words in this geometric vector space.

The above example assumes the skip-gram model. For the Continuous bag of words (CBOW), we would basically be predicting a word given the context.

GloVe:

GloVe is nearly like Word2Vec. Because Word2Vec is a "predictive" model predicting a particular context of words, GLOVE learns by constructing a matrix of co-occurrence (words X context) that basically counts when a word appears in a context [7]. Because it will be a giant matrix, to achieve a diminished representation of dimensions, this matrix is factorized. GLOVE has a lot of news, but that's the hard concept.

FastText:

FastText is very distinct from the embedding of the above two. While each word is treated by Word2Vec and GLOVE as the lowest unit to train, FastText utilizes n-gram characters as the lowest unit.[8] For instance, the word vector "apple" could be split into distinct units of word vectors as "ap”, "app”, "ple". The greatest advantage of using FastText is that it generates better word embedding for unusual words, or even words that are not seen during practice as the character vectors of n-gram are shared with other words. This can't be achieved by Word2Vec and GLOVE.

The steps used for preprocessing our database is as below,

We need to get the vocabulary as close as we can to the embedding. GoogleNews Word2Vec pretrained embeddings is being used for the word vectorization.

Next a function is defined that checks the intersection between our vocabulary and the embeddings. It will output a list of out of vocabulary words that can be used to improve the preprocessing.

As only 24% of the databases vocabulary will have embeddings, it makes 21% of our data more or less useless. A look at the top out of vocabulary words,

```
[('bitcoin', 987),
 ('Quorans', 858),
 ('cryptocurrency', 822),
 ('Snapchat', 807),
 ('btech', 632),
 ('Brexit', 493),
 ('cryptocurrencies', 481),
 ('blockchain', 474),
 ('behaviour', 468),
 ('upvotes', 432)]
```

Figure 8: Top out of vocabulary words (1)

For the punctuation, the question is should it be considered it as a token or not, so for the punctuations, If the token has an embedding, it stays, else it is removed. So, let's check:

So basically, a function is defined that splits off "&" and removes other punctuation.

After this function, embeddings ratio increases from 24% to 57% by just handling punctuation. Now another look at the top 1o out of vocabulary words,

```
[('to', 406298),
 ('a', 403852),
 ('of', 332964),
 ('and', 254081),
 ('2017', 8781),
 ('2018', 7373),
 ('10', 6642),
 ('12', 3694),
 ('20', 2942),
 ('100', 2883)]
```

Figure 9: Top out of vocabulary words (2)

These words prove that numbers also are a problem. The top 1o embeddings at this point are,

```
</s>
in
for
that
is
on
##
The
with
said
```

Figure 10: Top 10 embeddings

The '##' is there simply because as a reprocessing all numbers bigger than 9 have been replaced by hashes. I.e. 15 becomes ## while 123 becomes ### or 15.8o€ becomes ##.##€. So, this preprocessing step is mimicked to further improve the embeddings coverage, that is another 3% increase.

So, the common misspellings are checked out when using American/ British vocab and replacing a few "modern" words with "social media" for this task a multi regex script was used. Additionally, the words "a", "to", "and" and "of" are simply removed since those have obviously been down sampled when training the GoogleNews Embeddings.

The top out of vocabulary words now,

```
[('bitcoin', 987),
 ('Quorans', 858),
 ('cryptocurrency', 822),
 ('Snapchat', 807),
 ('btech', 632),
 ('Brexit', 493),
 ('cryptocurrencies', 481),
 ('blockchain', 474),
 ('behaviour', 468),
 ('upvotes', 432),
 ('programme', 402),
 ('Redmi', 379),
 ('realise', 371),
 ('defence', 364),
 ('KVPY', 349),
 ('Paytm', 334),
 ('grey', 299),
 ('mtech', 281),
 ('Btech', 262),
 ('bitcoins', 254)]
```

Figure 11: Top out of vocabulary words (3)

Now there are no obvious out of vocabulary words there to be quickly fixed.

CHAPTER 3

PROCESSING

3.1 MACHINE LEARNING GENERAL

Machine learning is a data analysis method which teaches computers to do what comes so naturally to animals and people: to learn through experience. Machine learning algorithms use computer techniques to directly "learn" information without depending on a predefined equation as a model. As the number of samples accessible for learning increases, their effectiveness is enhanced by the algorithms.

The greater part of these applications can be depicted inside the standard machine learning work process, which includes four stages: information cleaning and pre-processing, highlight extraction, demonstrate fitting and assessment (Figure 3) [9]. It is standard to signify one information test, including all covariates and highlights as info x (more often than not a vector of numbers), and mark it with its reaction variable or yield esteem y (as a rule single number) when accessible

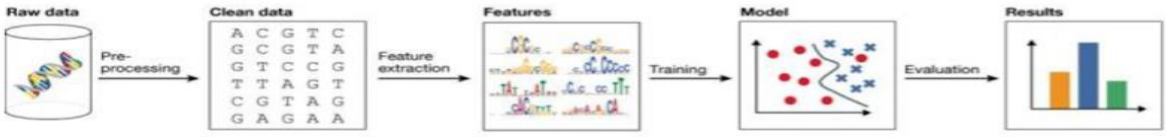


Figure 12: The classical machine learning workflow can be broken down into four steps: data processing, feature extraction, model learning and model evaluation

A supervised model of machine learning seeks at learning a feature f(x) = y from a list of training pairs (x 1, y 1), (x 2, y 2), ... for which information is registered. The x inputs, computed from the raw information, reflect what the model "reads about the world" and are extremely problem-specific in their selection. It is crucial for performance to derive most

informative characteristics, but the method can be labor-intensive and needs understanding of the domain. This limiting factor is particularly restrictive for high-dimensional information; even computational selection techniques do not measure the usefulness of the large amount of possible input combinations. A significant latest advance in machine learning is to automate this critical step by learning from profound artificial neural networks an appropriate representation of information. In short, a profound neural network requires raw information to the smallest (input) layer and converts it into increasingly abstract depictions of features by successively merging data-driven outputs from the previous layer, encapsulating extremely complex tasks in the process. Deep learning is now one of the most active areas of machine learning, and it has been shown to enhance picture and speech recognition efficiency, comprehension of natural language, and most lately computational biology.

3.1.1 LSTM- Long Short-Term Memory

Long short-term memory is a recurring deep learning architecture that uses memory cell vectors and a set of element multiplication gates to control the way information is stored, forgotten, and utilized in the network [10]. Long Short-Term Memory Networks, generally referred to as "LSTMs"–are a unique type of RNN that can learn long-term dependencies [11]. Hochreiter & Schmidhuber introduced them and many individuals refined and popularized them in following research. They operate tremendously well on a wide range of issues and are now commonly used. LSTMs are specifically designed to avoid the long-term dependence problem. Their default behavior is to remember long-term information, not something they're trying to know! All recurring neural networks form a chain of repeating modules of the neural network. This repeating module will have a very simple structure in standard RNNs, like a single tanh layer.

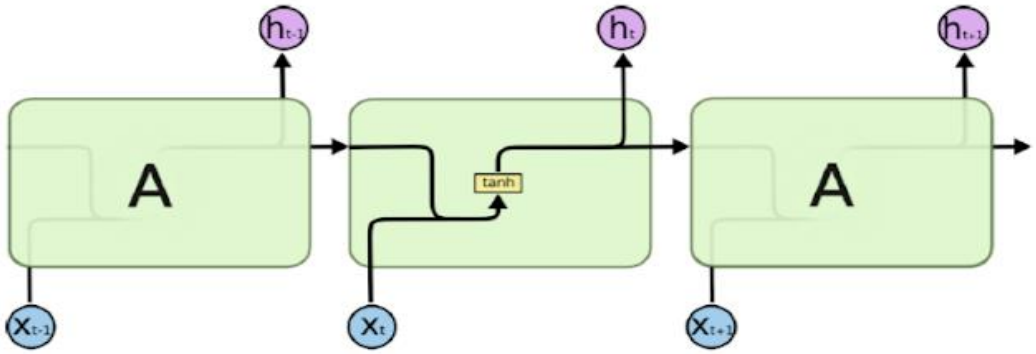


Figure 13: The repeating module in a standard RNN contains a single layer

LSTMs also have this chain as a framework, but there is a distinct structure in the reiterating module. There are four, rather than having a single layer of neural network, communicating in a very unique manner[12].

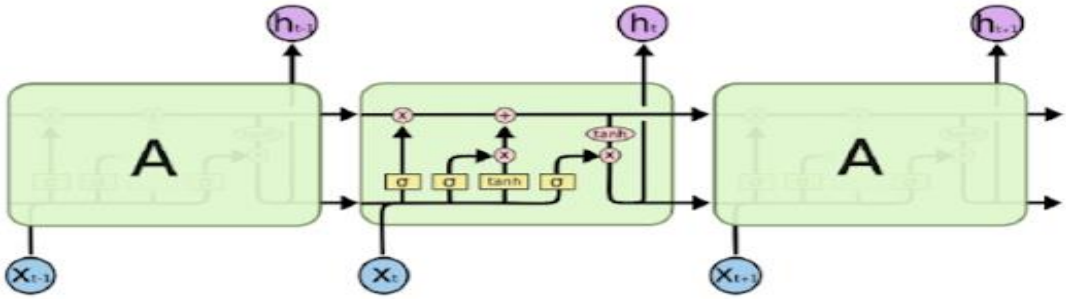


Figure 14: The repeating module in an LSTM contains four interacting layers

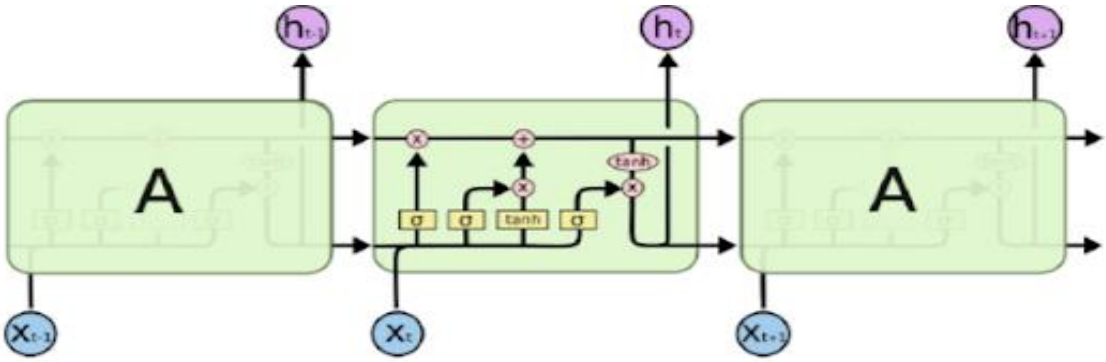


Figure 15: Notations explained

In the diagram above, each row carries a whole vector, from one node's output to others ' outputs. The purple circles are point-sensitive operations, such as adding vectors, while the yellow boxes are learned parts of the neural network. Merging lines indicate concatenation, while a line forking denotes copying its material and copying it to distinct places.

A fundamental sequence-to-sequence model comprises of two recurrent neural networks (RNNs): an encoder processing the input and an output generating decoder. Below is a description of this fundamental architecture.

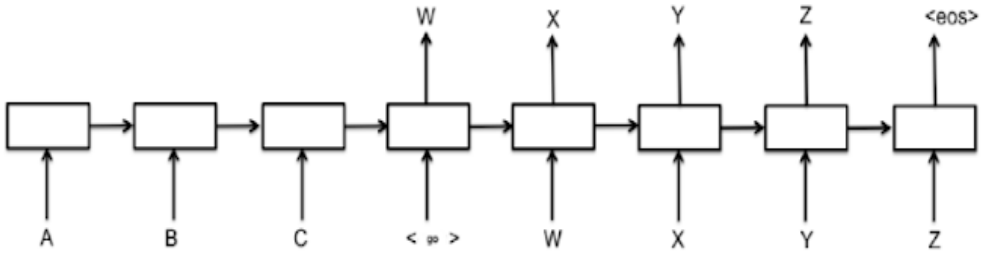


Figure 16: Typical Seq2Seq architecture

Each box in the above image reflects an RNN cell, usually a GRU cell or a LSTM cell. Encoder and decoder may share weights or use a distinct set of parameters, as is more prevalent [13]. Also, in sequence-to-sequence models, e.g. for translation, multi-layer cells were used effectively.

Each input must be encoded into a fixed-size state vector in the basic model shown above, as that is the only thing carried to the decoder. A mechanism of attention was implemented in order to enable the decoder to have more immediate access to the input. At each decoding phase, it enables the decoder to look into the input. This looks like a multi-layer sequence-to-sequence network with LSTM cells and the decoder's attention mechanism.

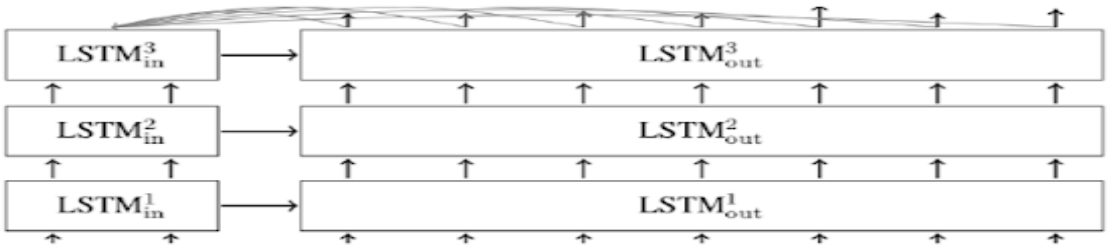
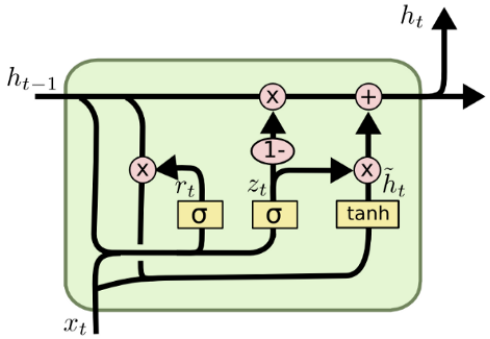


Figure 17: Multi-layer seq2seq network with LSTM cells and attention mechanism

3.1.2 GRU-Gated Recurrent Unit



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Figure 18: GRU basic structure

Introduced in 2014 by Cho, et al GRU (Gated Recurrent Unit), similar to LSTM but simpler to compute [14], aimed at solving the issue of the disappearing gradient that comes with a standard recurring neural network. GRU can also be regarded as a variation on the LSTM because both are likewise constructed and generate equally great outcomes in some instances.

A GRU instead of having a simple neural network with four nodes as the RNN had previously has a cell containing multiple operations (green box in the figure). Now the model

that is being repeated every sequence is the green box containing three models (yellow boxes) where each one of those could be a neural network.

GRU utilizes the gate known as, reset, and update. These gates are represented by the Sigma notation above: allowing a GRU to carry data over many periods of time to impact a future period of time. In other words, for a certain amount of time, the value is located in memory and at a critical point attempting to pull that value out and using it to modify at a future date with the current state.

Basic components of a GRU

1. Update gate

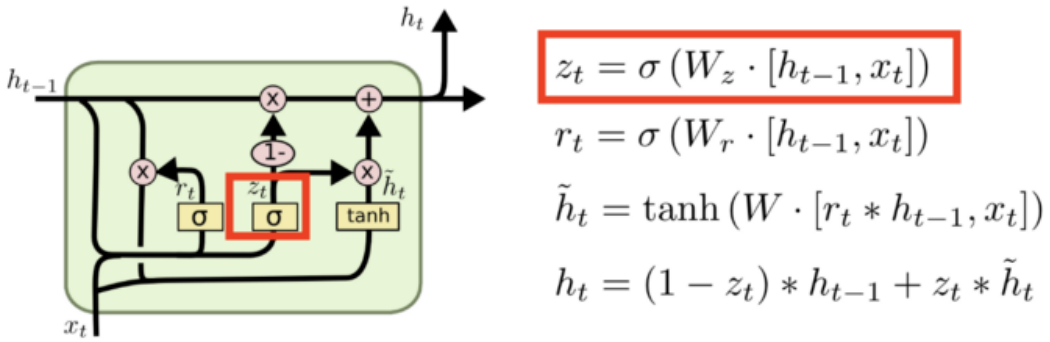


Figure 19: Update Gate

This gate calculates the update gate Zt at time step t executing the following steps:

- The input Xt is multiplied by a weight Wzx
- The previous output Ht-1 which hold information from previous units multiplied by a weight Wzh
- Both are added together and to squeeze output between 0 and 1 a sigmoid function is applied.

Given this gate the issue of the vanishing gradient is eliminated since the model on its own learn how much of the past information to pass to the future.

2. Reset gate

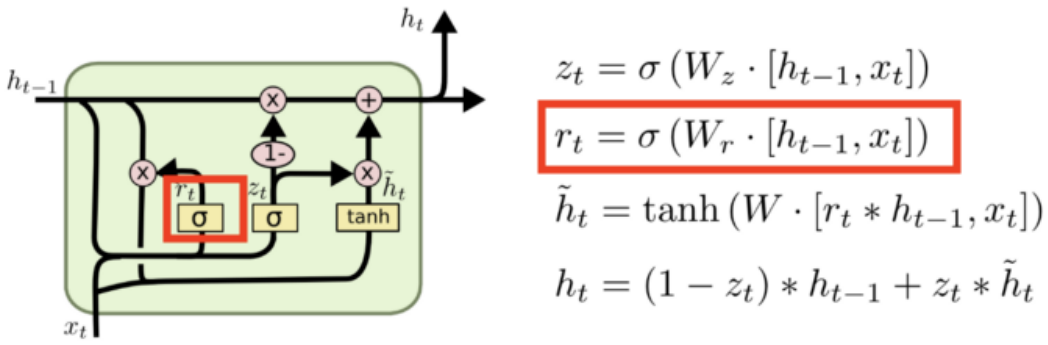


Figure 20: Reset Gate

This gate calculates the update gate Rt at time step t executing the following steps:

- The input Xt is multiplied by a weight Wrx
- The previous output Ht-1 which hold information from previous units multiplied by a weight Wrh
- Both are added together and to squeeze output between 0 and 1 a sigmoid function is applied.

The formula is very similar with the above gate and differs only in the weights and the gate's usage.

This gate has the opposite functionality in comparison with the update gate since it is used by the model to decide how much of the past information to forget.

3. Current memory content

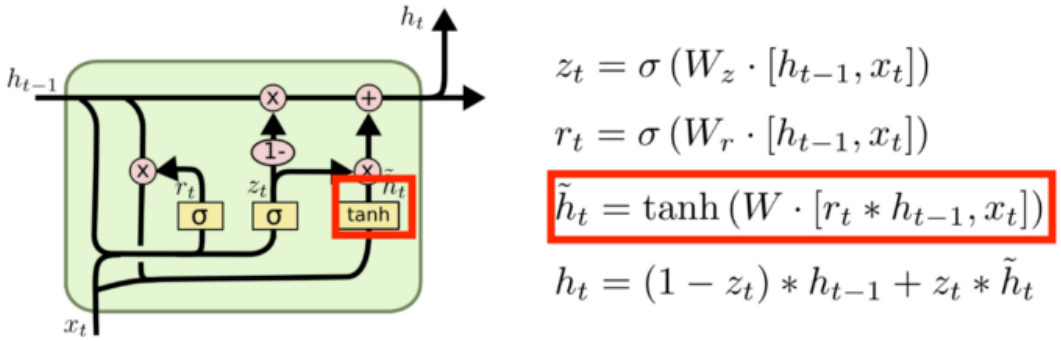


Figure 21: Current memory content

The calculation involves the following steps:

- The input Xt is multiplied by a weight Wx
- Apply element-wise multiplication to the reset gate R*t* and the previous output Ht-1; this allows to pass only the relevant past information.
- Both are added together and a tanh function is applied.
-

4. Final memory at current time step

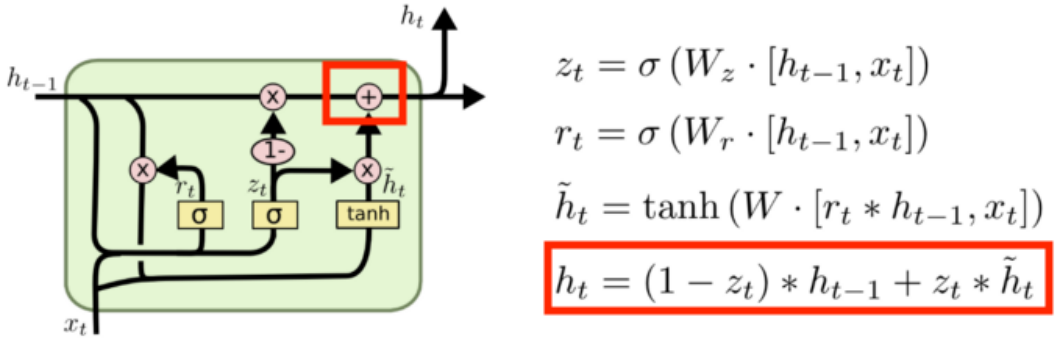


Figure 22: Final memory at current time step

Last but not least the unit has to calculate the Ht vector which holds information for the current unit and it will pass it further down to the network. Key role in this process plays the update gate Zt. [15]

The calculation involves the following steps:

- Apply element-wise multiplication to the update gate Zt and *Ht.*
- Apply element-wise multiplication to one minus the update gate 1-*Z*t and H'*t.*
- Both are added together.

If the vector Zt is close to 0 a big portion of the current content will be ignored since it is irrelevant for our prediction. At the same time since *Zt* will be close to 0 at this time step, *1-Zt* will be close to 1 allowing the majority of the past information to be kept. For example, we may predict the Saturday sales of a shop and probably we weight more the sales of last Saturday in comparison with yesterday sales (Friday's).

GRUs using the internal memory capability are valuable to store and filter information using their update and reset gates. That said the issues faced by RNNs (vanishing gradient problem) are eliminated offering us a powerful tool to handle sequence data.

3.2 EVALUATION METHOD

The evaluation metric for the models is the F1 score (the harmonic average of precision and recall), which penalizes the model's approval of insincere questions and flagging of sincere ones.

3.2.1 Precision

Precision is the number of true positives divided by all positive predictions. Precision also is termed Positive Predictive Value. It is a measure of a classifier's specificity. Low precision indicates a high number of false positives.

3.2.2 Recall

Recall is the number of true positives divided by the number of positive values in the test data. Recall is often termed Sensitivity or the True Positive Rate. It is a measure of a classifier's completeness. Low recall implies a high number of false negatives.

3.2.3 Trade-off between precision and recall

Precision implies the acceptable percentage of your outcomes. Recall, on the other hand, relates to the proportion of complete acceptable outcomes that your algorithm properly classifies.

We will need to consider both of these measures when constructing a classification model. Trade-off curves comparable to the graph below are typical for the review of classification models associated metrics. The thing to remember is that you can adjust the model to be anywhere along the border.

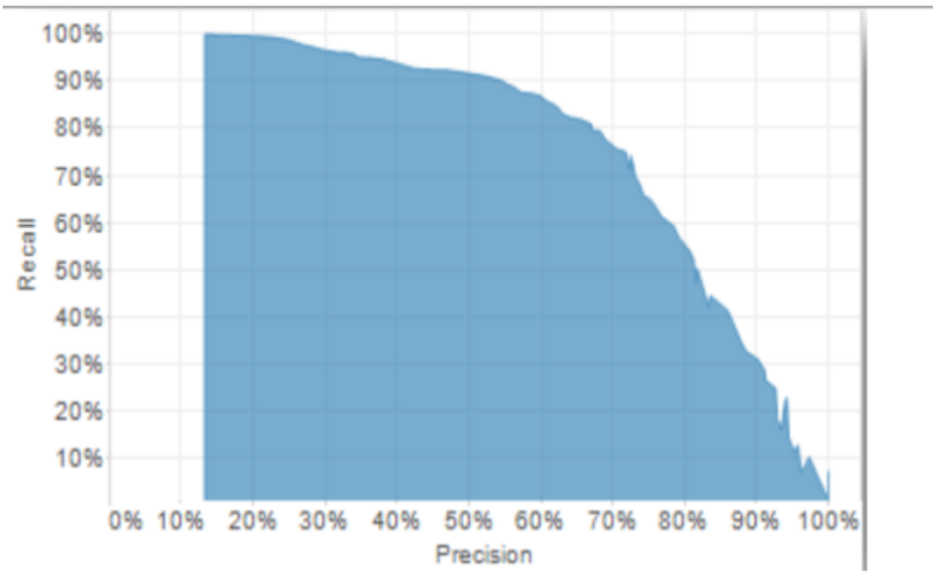


Figure 23: Precision and Recall trade-off

For a given model, either statistic can always be increased at the expense of the other. Choosing the desired mix of precision and recall can be regarded equivalent to turning a dial between projections that are more or less conservative (i.e. recall-focused versus precision-focused). It is essential to note that this is for a particular model; in reality, a better model can improve both precision and recall.

In choosing the correct balance of precision and recall, you should carefully consider the problem you want to solve. This competition uses the F1 score (the harmonic average of precision and recall), which penalizes the model's approval of insincere questions and flagging of sincere ones and balances precision and recall.

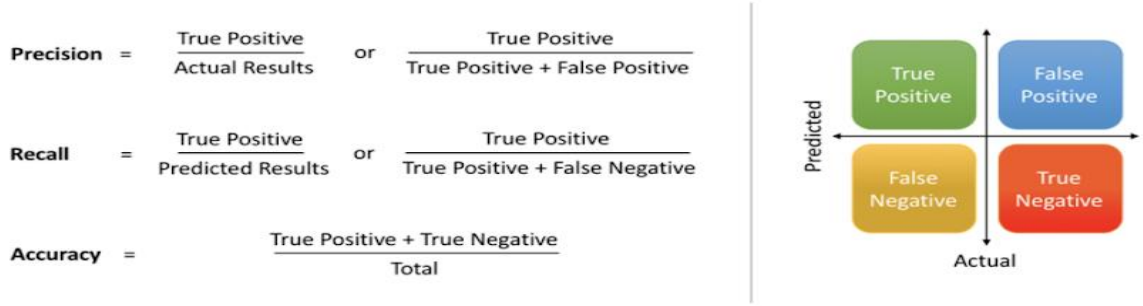


Figure 24: Precision and Recall

CHAPTER 4

RESULTS

Three different variations of LSTM with embeddings was run on the Dataset and the results are compiled in the table below.

Model Type	Accuracy	F1 Score
LSTM	0.954	0.77
Bidirectional LSTM	0.9603	0.682
LSTM+GRU	0.959	0.676

Table 4: Results

CHAPTER 5

CONCLUSION

This project focuses on the detection of toxic contents of Quora questions using methods of machine learning, which is a Kaggle competition. After attempting different models and parameter adjustment combinations, it was discovered that using word embedding in the information processing era and using Bidirectional LSTM to suit the information produced the highest output. LSTM works well with Natural Language processing problems.

As the dataset was pretty huge several iterations were required to run the algorithm. Embeddings turned out to be as a good boost for the dataset, as it performed better. For future work, add' attention' to the LSTM and suggest a Long Short-Term Memory Network based on Attention for classification of aspect-level sentiments. When different aspects are taken as input, the attention mechanism can focus on different parts of a sentence. Convolutional neural networks can be used as a recurring structure to capture contextual information as far as possible, resulting in significantly less noise compared to traditional neural networks based on windows.

REFERENCES

[1] Craig Smith. 2019. Interesting Facts and Statistics About Quora. https:// expandedramblings.com/index.php/quora-statistics

[2] Kaggle Inc. 2019. Quora Insincere Questions Classification: Detect toxic content to improve online conversations. https://www.kaggle.com/c/ Quora-insincere-questions-classification

[3] A Novel Evolutionary Preprocessing Method Based on Over-sampling and Under-sampling for Imbalanced Datasets https://ieeexplore.ieee.org/document/6699499

[4] Intent Detection using Semantically Enriched Word Embeddings https://ieeexplore.ieee.org/document/7846297

[5] Encoder-Decoder with focus mechanism for sequence labelling based spoken language understanding, http://docplayer.net/50706083-Arxiv-v2-cs-cl-13-mar-2017.html

[6] Sentiment Analysis of Bengali Comments With Word2Vec and Sentiment Information of Words https://ieeexplore.ieee.org/document/7912903

[7] Vector representation of words for sentiment analysis using GloVe https://ieeexplore.ieee.org/document/8324059

[8] Automation in Social Networking Comments With the Help of Robust fastText and CNN https://ieeexplore.ieee.org/document/8741503

[9] Deep learning for computational biology, https://www.turnitin.com/t_inbox.asp?aid=78671491&svr=56&session-id=c94acf0c5bfd29c46712b8f8c5a0de90&lang=en_int&r=0.5859834766752379

[10] Using LSTM and GRU neural networks method for traffic flow prediction https://ieeexplore.ieee.org/document/7804912

[11] LSTM Easy-first Dependency Parsing with Pre-trained Word Embeddings and Character-level Word Embeddings in Vietnamese https://ieeexplore.ieee.org/document/8573397

[12] An Improved LSTM Structure for Natural Language Processing https://ieeexplore.ieee.org/document/8690387

[13] Refining Word Embeddings Using Intensity Scores for Sentiment Analysis https://ieeexplore.ieee.org/document/8241844

[14] Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation, Kyunghyun Cho, Dzmitry Bahdanau, https://arxiv.org/pdf/1406.1078v3.pdf

[15] ET-GRU: using multi-layer gated recurrent units to identify electron transport proteins, https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-019-2972-