Votes Prediction In StackExchange Answers

Aqeel Labash
Faculty of Science and Technology
University of Tartu
Tartu , Estonia
aqeel.labash@gmail.com
Supervisor
Tambet Matiisen

ABSTRACT

The aim of this project is to see if we can truly predict number of votes a StackExchange answer can get. That is depending only on the question and the answer text using LSTM [1] network.

CCS Concepts

•**Computing methodologies** → **Neural networks;** *Natural language processing;*

Keywords

Deep Learning; StackExchange; votes prediction; Long Short Term Memory

1. INTRODUCTION

Every day around 10-20M [2] views, 4-10M users visit StackOverflow website [1]. Every minute 4.6 answers,2.84 questions. With 11,573,980 questions and 18,713,658 answers, 5,509,974 users ,56,372,889 comments.[2] This huge amount of data exists on StackOverflow website. StackOverflow is only one part of StackExchange. Where people can post questions and they are answered by the users. It has a feature which is, each answer can get votes based on the quality of the answer. Consequently, Voting is the vital factor that determines how good is an answer to a specific question.Which allow raising the web site quality without human interaction. Therefore, being able to predict the expected number of votes before even submitting it is the major motivation for this project.

2. BACKGROUND

2.1 Word Embedding

[1]Long-Short-Term-Memory
[2]Million

DOI: 10.475/123_4

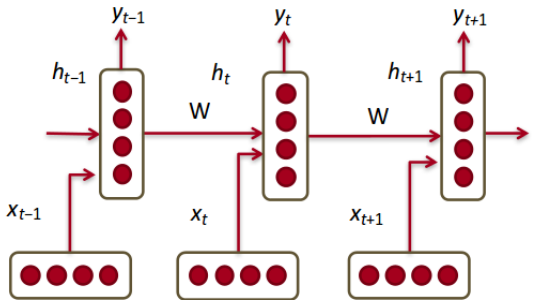

Figure 1: Recurrent Neural Network [13]

It's about mapping words to vectors of real numbers where the dimension of that vector is very low compared to vocabulary size.

$$W : words \rightarrow \mathbb{R}^n [14]$$

The resulted vectors can show semantic-syntactic releationship between words. [12]
Find the mapping could be done using neural networks.[4]

2.2 Deep Learning

There are many definitions for deep learning. One of them is:"A class of machine learning techniques that exploit many layers of non-linear information processing for supervised or unsupervised feature extraction and transformation, and for pattern analysis and classification."[8] another definition explain it as an algorithms based sub-field in machine learning to learn multiple levels of representation in order to model complex relationship between features.[8] One of the methods used in deep learning is Recurrent Neural Networks.

2.3 Recurrent Neural Networks-RNN

It is a class of artificial neural network. It is renowned for it's dependency on previous information (sequences).[9] [10] Which traditional neural networks lack because they treat all inputs and outputs as independent from each other which is not suitable for all cases.[10] Specially when we are working with natural languages where each word depends on series of previous words.
In Figure 1 we can see that the result at certain time y_t depend on x_{t-1}, x_t.But it does not depend on x_{t-1} directly, weights have to be applied first.

2.4 Vanishing gradient and gradient explosion

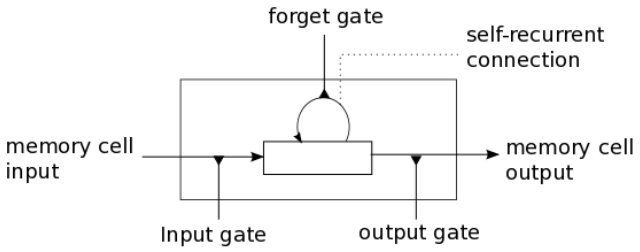

Figure 2: Memory cell [3]

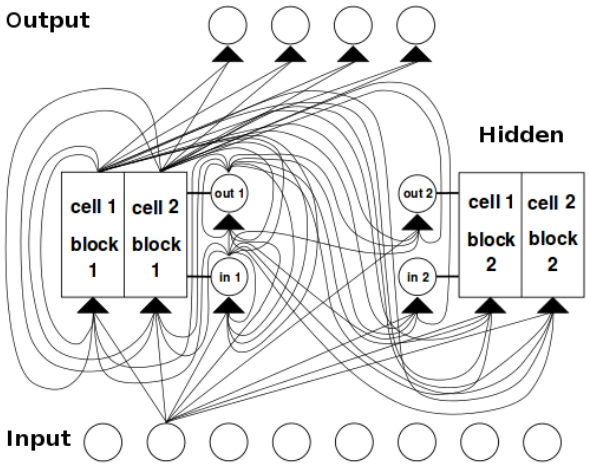

Figure 3: Example of a net with 2 memory cell blocks of size 2 [7]

In RNN, there are two issues widely known,gradient vanishing and gradient explosion.[15] The RNN propagate weights from time step to next time step.This property allows it to take the words context into consideration.In practice, RNN is more likely to less predict the next word from words it saw many steps away.[13] "This is because during the back-propagation phase, the contribution of gradient values gradually vanishes as they propagate to earlier time-steps."[13] It's also possible that the gradient grow extremely large.[13]

2.5 Long Short Term Memory

Long Short Term Memory is a Recurrent neural network that introduced memory cell. Memory cell prevents gradient vanishing and explosion by using 4 gates Shown in Fig 2.Those gates modulate environment-memory cell interaction.

Self-recurrent connection:it has a weight of 1.It's main task to keep memory cell state constant regardless of any interference.

Input gate: allow or prevent incoming signal from changing the memory cell state.

Output gate: responsible of allowing the memory cell to affect other neurons.

Forget gate: this gate inflect self-recurrent connection to allow or prevent it from remembering the previous state.

3. DATASET

The data set I used to optimize the parameters were from astronomy Stachexchange. It has around 4536 answers and 2738 answered questions. I picked this because it is not so large not to take a long time in processing and at the same time it's not very small. After optimizing the parameters on this data, the same parameters used on the large data set which was from programmers Stackexchange. Which contain 126776 answers.

3.1 Preprocessing

Before using the data It should be organized and cleaned to get a better result.

1. **Organize data:**In this level, the task was to get the data that we might use, from XML files to CSV file.
2. **Remove HTML tags:**When working with data from the web usually it contains HTML tags which were the first thing to do. For this, I used Regular expression library in python.
3. **Remove special characters:**The answers and questions contained newlines, quotes and many other special characters which needed to be removed to generate a valid CSV file.
4. **Remove stop words:**Stop words are words like "in, to, of, etc..".Stop words usually have a high frequency which leads for them to act as noise more than features. To achieve this task I used NLTK [3][5] library.
5. **Stem words:**Stemming words is meant to return words to the verbal noun. For example, the words (sing, singer, singing, sings) have the same verbal noun which is sing. Stem words help in decreasing the dictionary size.Porter algorithm [16] were used to stem the words in this project.
6. **Outliers:**Outliers are entries in data set with an extreme value.In this project, both data sets (astronomy,programmers)contained outliers. I'll discuss this topic more in details in the next topic with statistics.

3.2 Simple Statistics

Table 1 shows some basic statistics about votes in questions and answers before and after removing the outliers. We can see in Figures 4 and 5 more clearly the distribution of answers votes before and after removing the outliers in astronomy dataset. All answers with votes more than 3 standard deviations away from the mean value were removed. In total 77 records were deleted.

Figure 6 shows the distribution of answers over votes. where we can clearly see that most votes lay between 0-5 votes.

For programmers dataset we can see in Table 2 the votes measurements before and after removing outliers.In programmers dataset around 1445 were deleted.Figures 7 and 8 shows questions and answers votes before and after removing the outliers.

4. MODEL

4.1 Start Point

At the start the model looked as described in table 3. I used MSE [4][11] as a loss function After that I started to

[3]Natural Language ToolKit

[4]Mean Square Error

Table 1: Questions, answers votes measurements before and after removing outliers in astronomy data set

	Min	Median	Mean	Max
Questions vote	-8	3	4.82	66
Answers vote	-8	2	3.34	62
Questions words length	3	38	50	3095
Answers words length	3	86	117	2154
Vocabulary size : 60645				
After Removing outliers				
Questions vote	-8	3	4.51	66
Answers vote	-8	2	2.98	15
Questions words length	3	36	48	3095
Answers words length	3	83	112	1858
Vocabulary size : 38746				

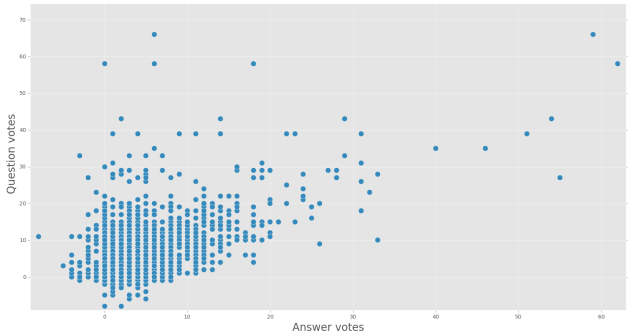

Figure 4: Shows questions votes vs answers votes before removing outliers in astronomy dataset

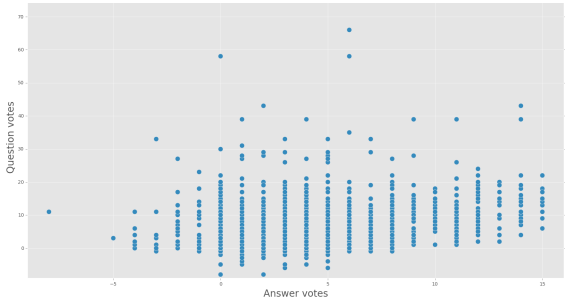

Figure 5: Shows questions votes vs answers votes after removing outliers in astronomy dataset

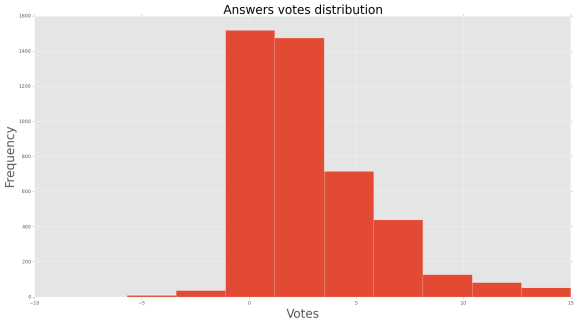

Figure 6: Shows answers distriution over votes for astronomy dataset

Table 2: Questions, answers votes measurements before and after removing outliers in programmers data set

	Min	Median	Mean	Max
Questions vote	-7	7	23.55	2189
Answers vote	-65	2	5.93	2402
Questions words length	2	69	89	2576
Answers words length	1	66	88	2422
Vocabulary size : 362094				
After Removing outliers				
Questions vote	-7	7	22.49	1050
Answers vote	-17	2	4.65	60
Questions words length	2	70	89	2576
Answers words length	1	65	87	2323
Vocabulary size : 358033				

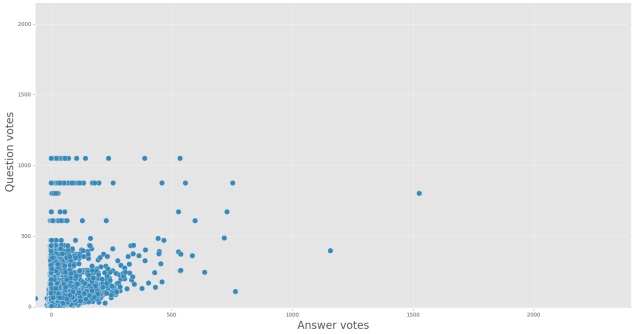

Figure 7: Shows questions votes vs answers votes before removing outliers in programmers dataset

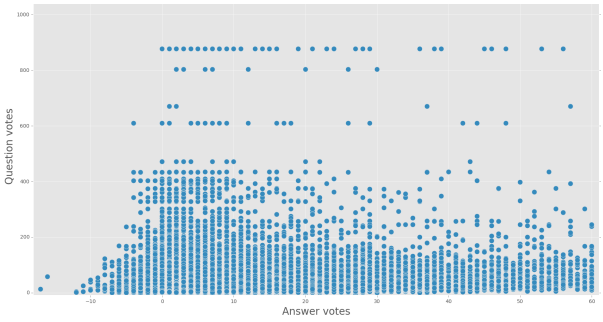

Figure 8: Shows questions votes vs answers votes after removing outliers in programmers dataset

Table 3: Layers details by order

Layer name	Parameter
Embedding Layer	100(hidden layer size)
Dropout layer	0.3(rate)
LSTM Layer	100(hidden layer size)
Dropout layer	0.3(rate)
Dense Layer	100(output dimension)
Dense Layer	100 (output dimension)
Dense Layer	1 (output dimension)

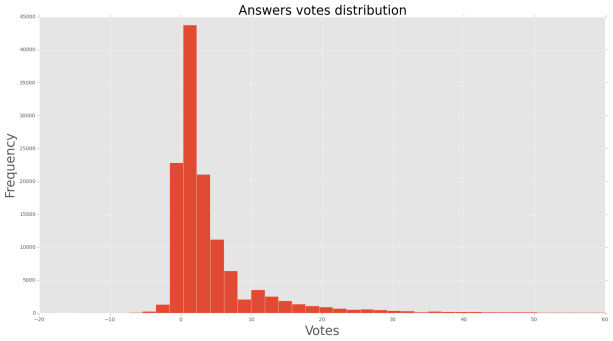

Figure 9: Shows answers distriution over votes for programmers dataset

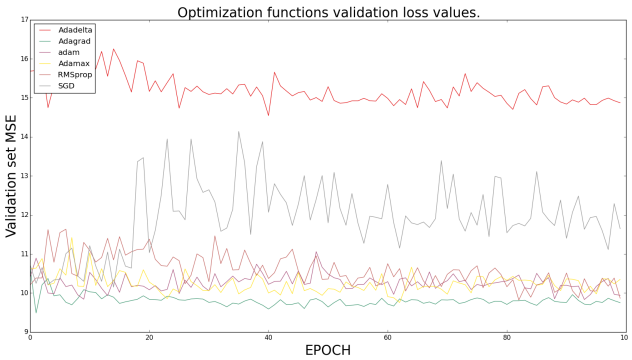

Figure 10: Shows MSE changes for validation set with epoch changes over many optimization functions.

optimize all the parameters step by step to reach the least MSE.
The data set was splited to two parts training : 80% and testing :20%. The training set was divided to give 5% as validation set.

4.2 EPOCH

Epoch number : is the number of times that all of training set used to update the weights. To decide the best number I depended on the validation loss. Figure 10 shows all the optimization functions[5] and the validation set MSE for epochs from 1-100. We can notice that Adagrad[6] perform the best at the begining and has better performance than other functions over all. If we take a look at training set MSE in Figure 11 we can see that Adagrad decrease a little bit more than the other functions. So dependong on those results I used Epoch :2 , and Adagrad as optimization function.

4.3 Question answer representation

To represent the data, I used dictionary index so each word would be weighted as a unique word.Using count representation for the words might lose the uniqueness of the words.
For the length of question and answer vectors, I used the mean value of words count.The increase in vectors length

[5]Keras optimizers
[6]adaptive gradient algorithm

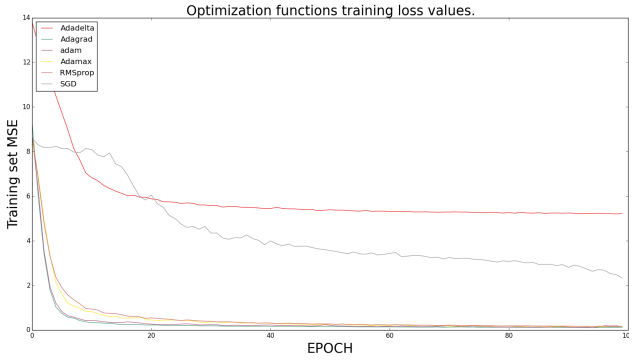

Figure 11: Shows MSE changes for training set with epoch changes over many optimization functions.

Table 4: Best loss function values for multiple embedding layer hidden size over validation set

Value	Best Validation Loss
25	9.7
50	10.69
75	10.13
100	9.48
125	9.65
150	10.19

will lead to the increase of MSE (Mean Square Error).After that, the result was fed to embedding layer.

4.4 Embedding Layer

This layer is used for words embedding. The questions and the answers were fed to seperate embedding layers. Deciding the size of hidden layers went through testing values. Table 4 shows the results I got.

4.5 LSTM Layer

The questions and answers were fed to two LSTM networks with the same hidden layer size. Testes been done to determine the hidden layer size that minimize the error. Table 5 shows the tests results.

4.6 Full connected layer

At the end, there is a fully connected layer that take as input the merged vector of questions and answers networks and output a single value that represent the prediction.

5. RESULTS

Table 5: Best loss function values for multiple LSTM layer hidden size over validation set

Value	Best Validation Loss
25	9.75
50	10.0013
75	9.97
100	9.48
125	10.18

5.1 Astronomy dataset

Astronomy data set size : 4460, train set size : 3568 , test set size : 892
After finalizing the parameters the lowest MSE achieved was 9.50 for testing set and 4.25 for training[7]. Here are some examples:
The answer that got the minimum error:
Question:[8] I want to get a tattoo of the solar system. The gap between the planets should not be preserved and the differences in sizes will only be symbolic. I found the following image My question is if all stars were to align to the same line as per the image above, would it show the same side of the planets as shown above? If not, where can I find a profile picture of which side of the planet it will show?
Answer: The planets aspects

Mercury surface is essentially a collection of small random craters with no discernible pattern at all, so you might not consider which side is presented. The only distinguished feature is a set of dark craters in its north pole.

Venus features few discernible aspects in visible light to the human eyes. There are only a few and faint distinguishable cloud bands, so you might also not consider which side is presented.

Earth is the most important, because it have continents and oceans with distinct designs. It also features a lot of clouds.

Mars has polar caps and a system of canyons. It's northern hemisphere is also much less cratered and has a lower altitude than its southern hemisphere, except that the greatest crater is in the southern hemisphere.

Jupiter has a banded structure of clouds covering the entire planet. It features a large red spot with some nearby fainter and smaller whiter spots.

Saturn also has a banded structure, which is faintier to Jupiter's structure but still clearly visible. It also features a curious hexagon on its north pole. But it is barely noticeable.

Uranus have very homogenous atmosphere (as seen in visible colors by human eyes), so it have almost no visible features to be drawn in your tatoo. Their presented side do not matters, because it is essentially a bland featureless ball.

Neptune has also few visible features. There are no more than a few cloud bands with low variation on color or hue. However there is a dark spot.[9] **Votes:**3
predicted votes:2.99

The answer that got the maximum error:
Question:[10] As I understand it, the asteroid belt exists because the gravitational force of Jupiter prevents the asteroids from accreting (is that a word?) into a planet.

If, however, Jupiter didn't exist and they did create a planet, how big would that planet be?
Answer: The mass of the asteroid main belt is estimated at 4% the mass of our moon according to Wikipedia so any object formed from the aggregation of that mass would not be a planet. It would be the size of a very small moon. Even if all the asteroids in the solar system were combined, the total mass would be below a third of the moon's mass.
Votes:15
Predicted votes:3.68

5.2 Programmers dataset

Programmers dataset size : 125329, train set size : 100263, test set size : 25066.
Using the same model used in astronomy dataset we got 54.59 for the testing set and for the training set 46.65. Here are some examples:
The answer that got the minimum error:
Question:[11]

I just finished watching this presentation by Uncle Bob (as well as his "Architecture" section of his "Clean Code" videos), but I'm left wondering:

Are there any examples out there of applications that implement this Entity-Boundary-Interactor (or Entity-Boundary-Controller) structure?

At one point I downloaded the source code to FitNesse (the acceptance testing project he mentions often as an example of not only high test coverage but good architecture, since they were able to defer the decision to not use a database until the very end), and based on a quick glance of it it appears even this project doesn't seem to fit this pattern.

Are there any nontrivial examples of this architecture out in the wild, or should I not bother even looking into it and chalk it up as "it would be great if you could get there, but nobody really does"?
Answer:I've not watched the video that you've linked, but I think I saw the same presentation a couple of months ago. Although the terminology is different, I believe the various components map pretty much precisely to the components you'd find in a typical CQRS architected DDD application. Certainly what Bob described in the presentation I saw matched the main project I work on. **Votes:**3 (currently 4) **predicted votes:**2.99

The answer that got the maximum error:
Question:[12]

I wrote an application that helps you to save energy. Actually it is very simple. I check the current location of the phone and I make some changes to the configuration like "sound off, dark display, wifi off...", depending on the location of the user. Sony just released a new phone including one of my apps features (actually they have an extra entry in the options menue for this). I have no idea whether there is a patent for this function.

Can I even release this app without risking to be sued some day?

I'm very confused about the whole "patent" situation. I'm about 20 years old and I can't even write a simple app, without investing lots of money for a lawyer.

Edit: I don't ask for legal advice. I wanted to receive an overview about how developers see or handle the whole situation.
Answer:

I am not a lawyer. There's a special word for people who

[7]If we continue we can minimize the training set error but there is no use if the testing error is increasing
[8]Best planets profile for a tattoo of the solar system
[9]Too long answer please continue at the link of the question
[10]How big would the asteroid belt planet be?
[11]Are There Any Examples of Uncle Bob's High-Falutin' Architecture?
[12]I want to publish an android app, but I'm afraid of software patents

take anonymous legal advice from the Internet - "fool".

Do a risk analysis -

a) you don't write the software.

outcome: Nothing.

b) you do write the software.

outcome #1: Sony doesn't notice and/or doesn't care. This might be a case of the "shallow pockets" defense - in their eyes you're not worth the effort to sue.

outcome #2: Sony sees your work and is impressed by it. They may offer to purchase it from you. (That might be cheaper than suing you then coding it up themselves)

outcome #3: Sony sees your work as an infringement that they must prevent. Step #1 (in the US, not sure about other countries) would be to send you a "cease & desist" letter. It's cheaper than suing, and chances are it's all they'd need to get you to stop.

Outcomes #1 and #2 are not harmful to you. Outcome #3 would probably mean you'd simply have to stop selling your software. You have to decide how likely #3 is and whether or not you can afford to stop.

My recommendation is to go ahead and write the software. You have much to gain (experience, reputation, possibly money) and little to lose. It's difficult to get the attention of large companies when you actually want it, so I'd expect outcome #1.

Good luck! **Votes:**60
predicted votes:2.08

6. DISCUSSION

Previously we mentioned that Adagrad got the best result at epoch 2 actually that mean that the model performs better at the start when the weights are randomized than after training. The problem cause might be related to two factors :

1. **Model:**One of the reasons might be that the model I used is not suitable for this kind of data or this kind of target.

2. **Features:**probably the features used is not enough for the model to generalize on other data.

3. **Data:** StackExchange working so hard to remove duplicates from the questions which would make every question-answer unique. Such thing would prevent the model from generalizing.

To understand how good or bad the model is performing, all predicted values were set to the mean value and then the MSE were calculated.For astronomy data set $MSE = 8.528692$ and for programmers data set $MSE = 57.65174$. The testing on astronomy dataset got $MSE : 9.50$ at the best case while the programmers dataset got $MSE : 54.59$. This mean that more training data would improve the results for sure. Figures 12, 13 shows the density of the predicted votes and the actual votes. depending on Figure 13 I think more training will make the prediction even better. And this oscillation in the prediction due to lack of training.

7. FUTURE WORK

There are many features that would decrease the error like:

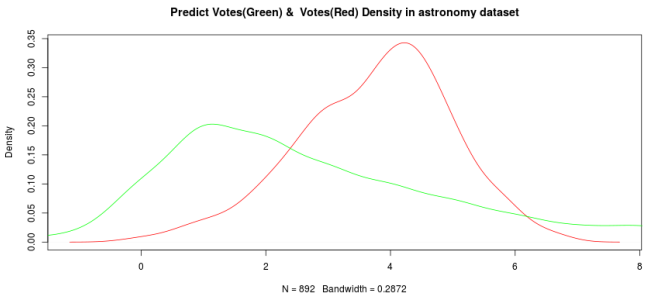

Figure 12: Density of predicted votes vs actual votes in astronomy dataset (testing set)

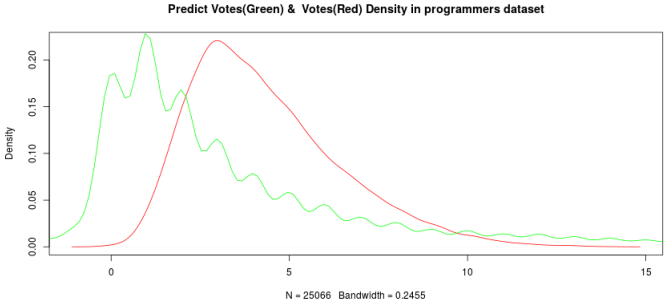

Figure 13: Density of predicted votes vs actual votes in programmers dataset(testing set)

1. Question view count: Which represent how much popular the topic is. Sometimes the answer is quite good but it's not a popular topic which leads to few votes.

2. Percentage between question view count and question score: This feature would give how good the question is.

3. Date : the older the post is the more chance it has to have more votes.

4. User reputation: the reputation of the user who answered the question have influence as well on the votes.Users with high reputation usually provide good answers.

5. User badges: each badge has a specific meaning which also can help in determining the quality of the user answers.

Use larger data set and try different StackExchange forums would help to optimize the model even more.But it'll need more resources and time.

8. CONCLUSION

In this project, StackExchange answers for astronomy dataset and programmers dataset were studied. The baseline was to beat the MSE when predicting answers votes. We used the astronomy dataset (due it's small size) to figure out the approximate settings for the model and then apply the result on the programmers dataset.
Although we couldn't beat the baseline in astronomy dataset but we did on programmers dataset. Depending on the current findings the prediction for the votes depending purely

on the text of the question and answer wouldn't be enough for accurate prediction. **Note:**All code,tex, .py,.ipython etc.. files available on Github

9. ACKNOWLEDGMENTS

The author of this report has been funded for his studies by IT Academy.
Keras: Deep Learning library for Theano and TensorFlow.[6]

10. REFERENCES

[1] https://api.stackexchange.com/docs/info. Online; accessed 2-May-2016.

[2] https://www.quantcast.com/stackoverflow.com. [Online; accessed 2-May-2016].

[3] http://deeplearning.net/tutorial/lstm.html.

[4] O. Barkan. Bayesian neural word embedding. *CoRR*, abs/1603.06571, 2016.

[5] E. L. Bird, Steven and E. Klein. Natural language processing with python, 2009.

[6] F. Chollet. keras. https://github.com/fchollet/keras, 2015.

[7] M. Creaney-Stockton and U. of Newcastle upon Tyne. Department of Electrical & Electronic Engineering. *Isolated Word Recognition Using Reduced Connectivity Neural Networks with Non-linear Time Alignment Methods*. University of Newcastle upon Tyne, 1996.

[8] L. Deng and D. Yu. Deep learning: Methods and applications. Technical Report MSR-TR-2014-21, May 2014.

[9] A. Graves, M. Liwicki, S. FernÃąndez, R. Bertolami, H. Bunke, and J. Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):855–868, May 2009.

[10] A. Karpathy. The unreasonable effectiveness of recurrent neural networks. Technical report, May 2015.

[11] E. L. C. Lehmann. George theory of point estimation. *Springer Texts in Statistics. Springer-Verlag*, 1998.

[12] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[13] R. S. Milad Mohammadi, Rohit Mundra. Deep learning for nlp, 2015.

[14] C. Olah. Deep learning, nlp, and representations. Technical report, July 2014.

[15] R. Pascanu, T. Mikolov, and Y. Bengio. Understanding the exploding gradient problem. *CoRR*, abs/1211.5063, 2012.

[16] K. Sparck Jones and P. Willett, editors. *Readings in Information Retrieval*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.