ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 19, April 2017

Answer ranking in community Q/A websites using Natural Language Processing

P.Sabitha[1], Tharun Kumar Reddy[2], B.Kali Das[3], V. Srinath[4]

[1] Asst. Professor in CSE Department, SRM University
[2]UG Scholar, CSE Department, SRM University
[3]UG Scholar, CSE Department, SRM University
[4]UG Scholar, CSE Department, SRM University
[1]sabithasanthosam11@gmail.com
[2]atkr323@gmail.com [3]srinathvuppula213@gmail.com [4]kalidas9849@gmail.com

Abstract: **Community blogs and Q/A websites have always been very helpful for the vast internet community in providing ideas, answers and suggestions to their various diversified questions, they range from technical, social, political, education etc. hence, providing the best results for the questions posed in the sites makes the lives of the people better and more efficient. This paper focusses on providing a new method for ranking the best answer for any given question posed in the community Q/A websites which is different from the traditional ranking based on sole point of ranking only with the help of number of votes that each answer obtains. Here in this paper we presented a new ranking method based on reviews/comments on the answer posts.**

Keywords: **Machine learning, Natural language processing.**

I. INTRODUCTION

Community Question Answering (CQA) sites have emerged in the past few years as an enormous market, so to speak, for the fulfillment of information needs. The CQA websites contains very detailed questions posted by the users and the community views the question, the informative users generally gives the answers descriptively instead of simply posting the referencing of the solution. The best answers among the dump of the answers will be ranked higher based on the number of votes it gets. The kind of questions which are posed in these websites doesn't generally contain a single line answer rather it requires a detailed answers which includes images , videos , referencing links etc hence , these kind of websites generally serve as a platform where ideas and solutions can be debated and discussed and the user who also gets visibility and encouragement which helps and drives other users to read/research the topic to present the answers to other questions.

Question answering (QA) helps one go beyond traditional keywords-based querying and retrieve information in more precise form than given by a document or a list of documents. Several community-based QA (CQA) services have emerged allowing information seekers pose their information need as questions and receive answers from their fellow users. A question may receive multiple answers from multiple users and the asker or the community can choose the best answer. While the asker can thus indicate if he was satisfied with the information he received, there is no clear way of evaluating the quality of that information.

In this paper we focus on delivering a new method of evaluating and ranking the best answer among the posted answers, in practical scenarios the ranking of the best answer will be based up on the number of votes a particular answer gets. The answer with the maximum number of votes will be ranked first and respectively for all the answers, but this ranking is not considering the suggestions and reviews a particular answer gets in their respective comments section hence, in our model we tried to device a method of evaluating a new score using the reviews written in the comment sections of the answer. This evaluation will be done using the machine learning algorithms and natural language processing techniques. This new method of evaluation gives new space for suggestions, ideas and productive suggestions and this method also paves a new way for the further improvement of this method by auto-tagging and automatic voting based on the text summarization which can be used to build effective AI based systems.

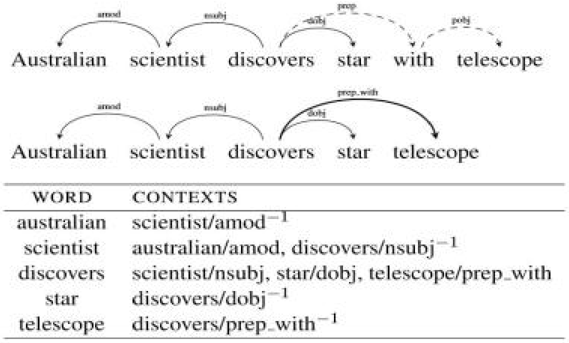
II. MACHINE LEARNING AND NATURAL LANGUAGE PROCESSING IN TEXT MINING.

Text mining is a process of deriving high quality content from a dump of data. High-quality information is typically derived through the devising of patterns and trends through means such as statistical pattern learning. Text mining usually involves the process of structuring the input text deriving patterns within the structured data, and finally evaluation and interpretation of the output. 'High quality' in text mining usually refers to some combination of relevance, novelty, and interestingness. Typical text mining tasks include text summarization and categorization, text clustering, and concept/entity extraction, production of granular taxonomies, sentiment analysis of the text dump , document summarization, and entity relation modeling.

Natural language processing is a part of machine learning that is effective in the text classification and text modelling. NLP's Sentiment analysis is widely applied to voice of the customer materials such as reviews and survey responses, online and social media, and healthcare materials for applications that range from marketing to customer service to clinical medicine which aims to determine the attitude of a speaker, writer, or other subject with respect to some topic or the overall contextual polarity or emotional reaction to a document, interaction, or event. The attitude may be a judgment or evaluation, affective state, or the intended emotional communication. here in this paper we use this method to analyze the comments of the users to classify the it into either of the three categories 1) positive 2) negative 3) neutral, the positive comment will and contribute in adding the additional score to the main score, negative comment will deduct the score and neutral comment will have no effect on the final score, but in this paper only those comments are considered as valid which are complete sentences which make sense.

Machine learning role in text mining, statistical analysis and pattern recognition is undeniable.it is used to identify the patterns using suitable algorithms, a trained machine learning model will be able to classify, analyze and summarize the data by becoming more and more efficient in each iteration of data input.

Basically here we analyze a particular statement and perform sentiment analysis on it, suppose if there exists a comment that says, "This is a wonderful post, it is very helpful" when we take each token in this statement and perform analysis on it, generally words that do not specify any meaning like "the"," is", "a" etc will be neglected and only nouns, verbs and adjectives will be considered. And all the valid words will be compared to a set of words in a dictionary. These words are classified in three categories 1) positive 2) negative 3) neutral when a statement is compared to any of these dictionaries the number of words that gets matches will be taken into account and classification will be made into either of these three categories based on number of matched words.



WORD	CONTEXTS
australian	scientist/amod^{-1}
scientist	australian/amod, discovers/nsubj^{-1}
discovers	scientist/nsubj, star/dobj, telescope/prep_with
star	discovers/dobj^{-1}
telescope	discovers/prep_with^{-1}

Fig 1.1- Automatic Text completion using NLP.

III.RANKING METHOD THROUGH TEXT MINING

Data Collection:

The data required for the ranking method includes the posts of the Q/A websites. We need the post's meta data such as the post ID, post's age, parent ID of the post, meta data of the comments and votes it gained. And using the API of the website in interest we can gather comments that each individual answer gets, and a cluster of the comments is made and the analysis is done on this cluster. Here, in this paper we focus on the community Q/A website Quora using the Quora API we can get the required data that we want for tour process, then feature engineering is performed. in this process we only select the required features that we want in the data that we have extracted.

Creating response variable:

For the response variable, we sought an objective metric of answer quality. We initially tried using pure score as our metric, but found that it was highly correlated to a variety of factors irrelevant to quality. For example, the natural log of score is correlated to age of the post (r = 0.31), a natural log estimate of the number of views of the answer post ln (parentViews * postAge/parentAge) (r = 0.24), age of the parent question post (r = 0.22), natural log of the ratio between post age and parent age in (postAge/parentAge) (r = 0.21), etc. This makes intuitive sense–an answer post which has existed for a longer time and has had more people view its original question post is likely to have a higher score regardless of quality.

For specific features, we chose post age, parent age, parent view counts, order of answer post, as well as each of their respective natural log transformations. Next, by running the ordinary least squares algorithm, we obtained adjusted scores (ynew = y−ˆ y) that were almost completely uncorrelated with these non-quality-related features (r = 0.00 for each), normally distributed around a mean of 0, and still leave a lot to be explained (R2 = 0.15 between y and ˆ y). Finally, we translate this value into a very simple and evenly split classification for each sample where helpful posts with score greater than mean are class 1, and unhelpful posts with score less than mean are class 0.

Pre-processing and Feature Selection

We have to consider taking all the question answer posts and metadata from the data dump, converting from xml to python dictionary format. Next, we created the following metadata and text-based feature matrices:

- **Topic tags:** log total and binary existence by type for parents (questions) of each answer post

- **XML tags:** log total, raw count and log count by type, score, and binary existence by type for each answer body text

- **Question metadata:** log # of characters, words, sentences, sum of question words, binary existence of question words

- **Answer metadata:** answer log number of characters, words, sentences

- **Word2vec representations:** Word2vec representation of each question and answer post's body text

- **Raw text data:** lower case unigram and bigram binary existence, raw and logged counts, scores for answer body

- **Q&A similarity:** Word2vec, cosine similarity between question and answer body text vectors • Interaction terms: 2nd degree interaction terms between a dimension-reduced matrix of word frequency features and a dimension-reduced matrix of topic metadata features

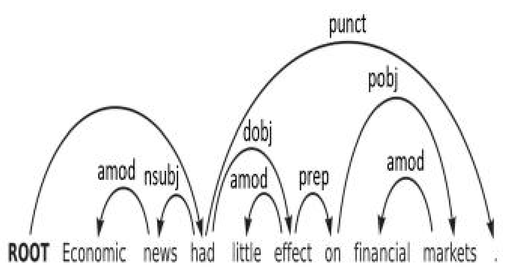


Fig 1.3- Parts of speech tagging in NLP to identify meaningful words for the text summarization process.

Implementations differ based on how they decide whether a given pair of words counts as a positive example. Traditional implementations use a context window of k words: if two words appear within k words of each other in the raw text, then the pair is a positive example. The dependency implementation considers a pair of words a positive example if one word is syntactically either the immediate head of or modifier of the other word (appearing either immediately above or below each other in the grammatical parse tree derived from the raw text). The advantage of this is that it goes beyond just capturing topical similarities between words to also removing spurious correlations between words that just happen to be close to each other in the text, as well as capturing functional similarities. Finally, to represent a given question/answer post, we take the average of the word2vec vectors of its individual words, effectively giving us a fixed, compact (300 variable) numerical representation for every post in our corpus. [4] discussed about Submerge Detection of Sensor Nodes. Underwater networking sensor nodes provide the oceanographic collection of data and monitoring of unmanned or autonomous underwater vehicle to explore sea recourses and gathering of scientific data. The sensor network contains the statistical data about the sensor nodes.

IV. PROBLEMS FACED

The most challenging problem that arises is due to the unintelligible comments posted by the users such as the repetitive words, posting symbols instead of complete comments, answers containing no comments at all etc are just some of the problems faced during the process, apart from this judging the answer based on the reviews/comments is a completely subjective topic as the review will be biased on one person's point of view.

V. PROPOSED SYSTEM

In this proposed system the trained model will be able to recognize the comment and will be able to classify into the any of the three categories and the summation of all the classification is taken into the account and score will be awarded to the answer similarly this process will be applied to all the answers for the question. And the ranking will be done based up on the final score obtained.

To evaluate our ranking model's performance, we treated it as an information retrieval (IR) system that seeks to return results with the good answers (class 1) being ranked (and thus appearing) before the bad ones (class 0), reflecting a real life situation where we would expect a community Q&A website to place the best results first on the page. Accuracy is then based on comparing the ranking our IR system returns to the ground truth ranking, with a focus on where the class 1 answers are located in the ranking. Especially for binary response variables, research literature commonly uses Mean

Average Precision (MAP) for this purpose, so we do so as well.

In order to carry out this evaluation, we needed a dataset organized by questions, where each question had at least two answers and these answers weren't all good or all bad.
We fit algorithms on the training set, used the validation set to choose hyper-parameters based on MAP performance, then averaged the optimal hyper-parameters obtained on each fold of cross validation, and applied them to the test set to get an estimated MAP.

VI. BENEFITS OF PROPOSED SOLUTION

The major benefit of this system will be distinguishing between a well-presented and a post with a good content. Generally in most of the posts the answer will be well presented with not–so-good content by good writers but as we consider quality in these community websites we would want to eliminate such posts and encourage the ones with the good content. With this new kind of scoring which is additionally presented along with the votes makes the posts which are far behind in ranking in preceding ranking method to come into limelight. And this also encourages in writing the answers with good content rather than presenting with poor content. The major benefits include

- Encourages active and productive debating for quality answers.
- Values reviews more than just normal voting
- More thinking space will be available to the viewer as he goes through all the suggestion that are recommended.

- Will be helpful in devising effective AI based systems.
- More research will be done by the writer before posting any answer.

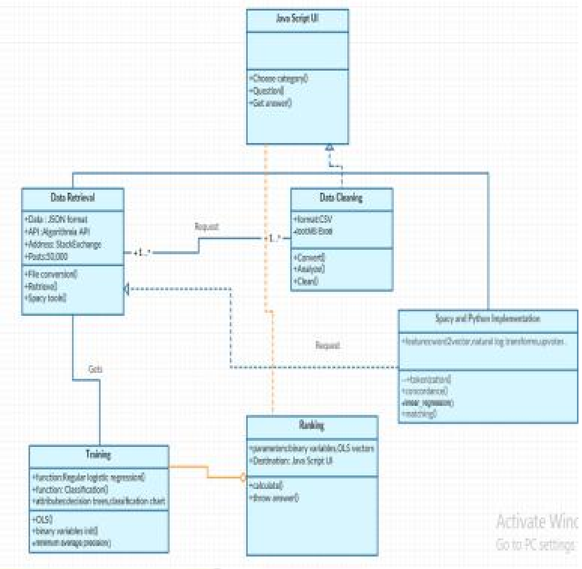


Fig-1.3-Architechture Diagram for the workflow in the ranking system.

VII. CONCLUSION

Currently our features are largely bag of words (not taking into account syntactic ordering), don't try to uncover semantic meaning, and are just based on the raw text itself, not using any third party authoritative source to determine whether posts accurately answer questions or not.

The further steps include auto tagging for each answer the AI based model on further improvisation made will be able to analyze the comment and convert into respective tags, and similarly this is done to all the comment where each comment will be converted in tags based on text summarization and tokenization techniques, and when two similar tags appear they count to one tag and the tag count will be incremented. Thus such kind of method paves a way for more intuitive judgment rather than simply voting and down voting an answer , this method also encourages active reviewing and open suggestions for the answer where the users will have more thinking space and coming to their conclusion.

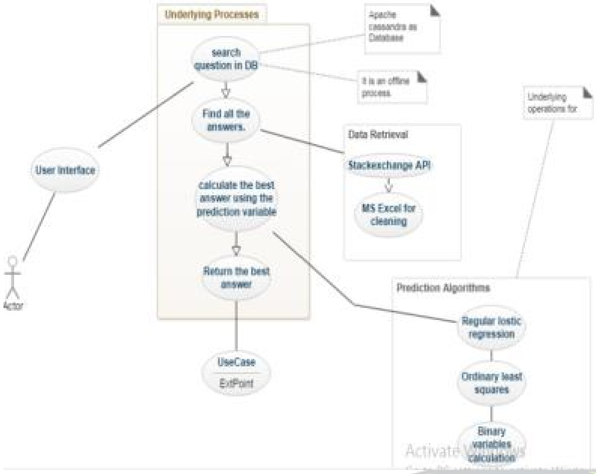


Fig 1.4-Use Case Diagram for underlying process of Answer ranking using NLP.

II. REFERENCES

[1] Ralf Herb rich, Thore Graepel, and Klaus Obermayer. “Large Margin Rank Boundaries for Ordinal Regression”. In: MIT Press, Jan. 2000. Chap. 7, pp. 115–132. url: http://research.microsoft.com/apps/pubs/default. aspx?id=65610.

[2] Lars Bentinck et al. “API design for machine learning software: experiences from the scikit-learn project”. In: ECML PKDD Workshop: Languages for Data Mining and Machine Learning.

[3] Sekine, S., Grishman, R., and Shinnou, H., A Decision Tree Method for Finding and Classifying Names in Texts, In Proceedings of the Sixth Workshop on Very Large Corpora, 171-178, Montreal, Canada, August, 1998.

[4] Christo Ananth, S.Surya, Berlin Mary, “Submerge Detection of Sensor Nodes”, International Journal Of Advanced Research Trends In Engineering And Technology (IJARTET), Volume II, Special Issue XXV, April 2015

[5] Voutilainen, A., Designing a (finite-state) parsing grammar, In Finite-State Language Processing, E. Roche and Y. Schabes (editors), A Bradford Book, The MIT Press, 1996.

[6] Grishman, R. and Sundheim, B., Message Understanding Conference-6: A Brief History. In Proceedings of the 16th International Conference on Computational Linguistics (COLING 96), 466-471, Copenhagen, August, 1996.

[7] Riseman, E.M. and Hanson, A.R., A contextual postprocessing system for error correction using binary n-grams, IEEE Transactions on Computers, C-23(5):480-493, 1974.