Research Article

# Intelligent Framework using Machine Learning Algorithms To RespondSocial Media Customers

**Priyanka Baviskar*[1], G.Vadivu[2]**
[1]PG Student, Big Data Analytics, SRM University, Tamilnadu, India.
[2] Professor, Department of Information Technology, SRM University, Tamilnadu, India.
***Corresponding author's E-mail:** baviskarpriyanka98@gmail.com

**Received:** 15-01-2017; **Revised:** 23-02-2017; **Accepted:** 11-03-2017.

**ABSTRACT**

The increasing number of people is using social media to express their personal experiences, reviews, and queries. This has led to an interest in supporting social media analysis for marketing, opinion analysis and understanding community cohesion. In this paper, we propose a social intelligence framework that can extract the questions asked via social media platform, like Twitter, to help the enterprise to understand more about the opinion of the customer towards the target product. Here, the opinion of the customer are the questions that are been asked by the customer related to the product. We have proposed an algorithm to extract and classify the customer questions from the twitter data, filter the questions into different product categories. This is used to develop a Lead Support Generation System which will help in answering all the questions asked by the customers without human intervention.

**Keywords:** Machine Learning, Naïve bayes Algorithm, Maximum entropy Algorithm, Classification, Python, Mongo DB, Hadoop.

## INTRODUCTION

Research in Question Answering (QA) seeks to move beyond the existing keyword-based Information Retrieval (IR) approaches by providing one or more *exact answers to a question* from a large document collection. The syntactic and semantic interpretation of a question is crucial in a QA system. The most common approach to semantic interpretation is to classify the question into a closed set of *question types like* Shoes, Model_merchendise, Games_Programme, Fan_question, Clothes, Video streaming, Enquiry_race, and Graphics_prints, which describes the expected semantic category of the answer to the question.

An important step in question answering (QA) system is to classify the question to the anticipated type of the answer. In this project we have proposed to extract the questions from twitter for testing purpose, the size of the sample twitter data is 10000. The Questions extracted from the tweets are related to the customer or the queries asked by the fan, through a tweet, e.g.: What is the location for F1 race this year? When is the match for German GP? Etc. The primary task after extracting the questions from the tweets is to categorize the questions into featured categories e.g.: Clothes, Collectable, Posters, Ticket Enquiry, and Live Streaming. For categorization of the questions, Machine learning algorithm is used. For this project we have proposed the usage of machine learning algorithm Naïve Bayes Algorithm Model.

Using the Naïve Bayes Algorithm a Naïve Bayes Classifier will be generated which will classify the questions into desired categories. The classifier will be the first feed with the training dataset. Here we have designed a training set of 1000 questions. 50 for each category for the above mentioned category of questions and passed it as training set to the classifier. After training the classifier we perform the Naïve Bayes Classification on the unknown instance of twitter question provides us with the probability with the most likely category of the question in which the question may fall into. The probability is arranged into the ascending order. Mongo DB is used to store the result set.

The categorized questions are stored as Questions Database using Mongo DB. From the stored questions Automatic response system is proposed which will generate an auto-response for the questions asked by the Customers/Fans, which is termed as Lead Support Generation.

### Literature Survey

Earlier question classification work by Pinto et al. (2002)[1] and Radev et at. (2002)[2], in which language model and Rappier rule learning were employed respectively. More recently, Li and Roth (2002)[3] have developed a machine learning approach which uses the SNoWlearning architecture (Khardon et al., 1999)[4]. They have compiled the UIUC question classification dataset which consists of 5500 training and 500 test questions. The questions in this dataset are collected from four sources: 4,500 English questions published by USC (Hovy et al., 2001)[5], about 500 manually constructed questions for a few rare classes, 894 TREC 8 and TREC 9 questions, and also 500 questions from TREC 10 which serve as the test dataset. All questions in the dataset have been manually labeled by them according to the course and fine grained categories, with coarse classes followed by their fine class refinements. In addition, they have considered the

distribution of the 500 test questions over such categories. Li and Roth (2002)[6] have made use of lexical words, part of speech tags, chunks (non-overlapping phrases), head chunks (the first noun chunk in a question) and named entities. They achieved 78.8% accuracy for 50 fine grained classes. With a hand built Dictionary of semantically related words, their system is able to reach 84.2%.

The UIUC dataset has laid a platform for the follow-up research. Hacioglu and Ward (2003)[7] used linear support vector machines with question word bigrams and error-correcting output to obtain accuracy of 80.2% to 82.0%. Zhang and Lee (2003)[8] used linear SVMs with all possible question word grams, and obtained accuracy of 79.2%. Later Li and Roth (2006)[9] used more semantic information sources including named entities, Word Net senses, class-specific related words, and distributional similarity based categories in question classification task. With all these semantic features plus the syntactic ones, their model was trained on 21'500 questions and was able to achieve the best accuracy of 89.3% on a test set of 1000 questions (taken from TREC 10 and TREC 11) for 50 fine classes. Most recently, Krishnan et al. (2005) used a short (typically one to three words) subsequence of question tokens as features for question classification. Their model can reach the accuracy of 86.2% using UIUC dataset over fine grained question categories, which is the highest reported accuracy on UIUC dataset.

In contrast to the previous approach we are using nltk's feature extraction approach for identifying the most prominent words of the question which will help the classifier to get the question into a particular category.

**Proposed work**

Purpose of this work is to extract the question asked in the tweets, categorize the Questions into different desired categories, and Store the Extracted questions into the Questions Database in the Mongo DB. To interpret the questions asked by the customer, and generate an automated response system; An automated response system will understand the question asked by the customer; interpret the question like a human, Match the question asked by the customer with the preexisting tweets , Extract answers from those tweets and give it as a response to the customer.

Store the responses with the respective query into a database for further references.

The questions extracted are divided into two main categories of questions: Questions asked by the fans or Questions asked by the customers. As the next step those questions are identified as Non-Wh questions and Wh-Questions. The categorization is performed with the help of regular expressions.

Naïve Bayes algorithm is used for this work, which is based on conditional probabilities. It uses Bayes' Theorem, a formula that calculates a probability by counting the frequency of values and combinations of values. Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred. If B represents the dependent event and A represents the prior event, Bayes' theorem can be stated as follows:

$P(H|X) = P(X|H)\, P(H)/P(X)$

Where,

- P (H|X) is the posterior probability, or a posterior probability, of H conditioned on X.
- P (H) is the prior probability, or apriori probability, of H.
- P(X|H) is the posterior probability of X conditioned on H

Naive Bayes algorithm in question classification proceeds by finding out the feature associated with each category. Features are the Headword occurring in questions pertaining to one particular category expressed in terms of their relevance in particular question.

These Headwords are mostly the category names or words associated with the category name. The Category names are passed as label to the classifier. Under each label there are 50 sample questions which will be the training set for the classifier.

After the classifier is trained the classifier will be applied on the test data, the Model is identified and accuracy will be calculated. If it is up to the desired accuracy then it will be implemented on larger data sets.

Further, to modify the categories into a detailed level, especially the fan questions. Each Fan question will be classified into a detailed category whether the customer is asking about the driver or about the race or about the grand prix. And also to generate an automatic response bot for answering the customer questions. In this paper, proposed few methods of responding to the customer queries. First, by specifying links and second by responding as stored static responses.

**Data Flow Diagram**

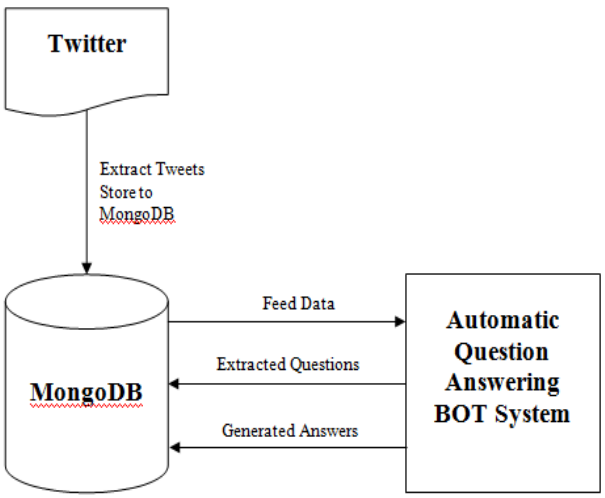


**Figure 1:** Data Flow Diagram

**System Architecture view**

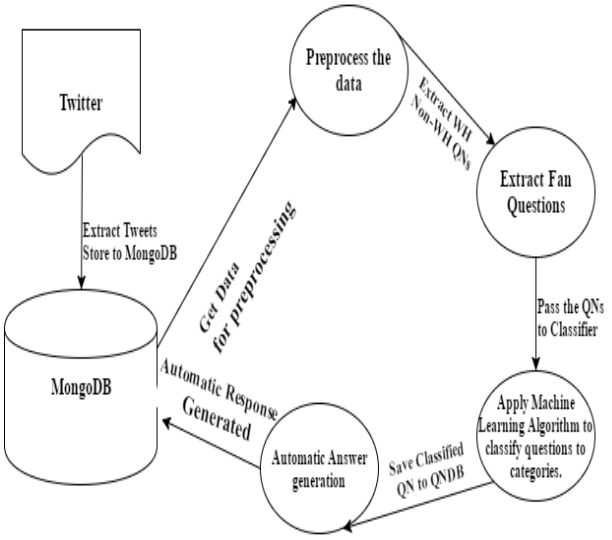


**Figure 2:** System Architecture view

**Flow Chart**

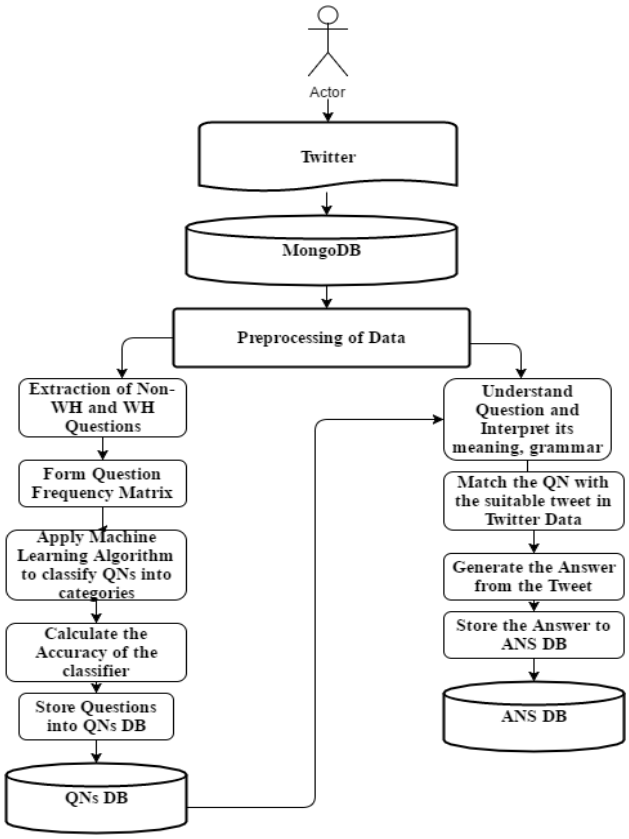


**Figure 3:** Flow Chart

**Generation of responses as links**

Get a customer query from the question DB, read the query and match it with the tweets in the tweets DB. Gather all the tweets which match the words from the queries and store them in a temporary variable. From these tweets capture all the links that are present inside the tweets, this can be done with the help of regular expressions in python. Finally store it as a response to the query into answer DB.

**Question classification mechanism**

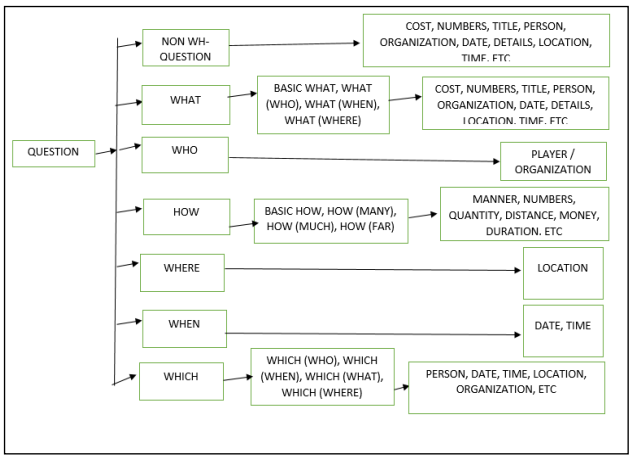


**Figure 4:** Question classification mechanism

**Generation of responses as stored static responses:**

In this work, created a collection of static frequently asked question and answer by the customers. If the Query matches the pre-existing query then response it with the pre-existing queries stored response.

## RESULTS

The output given below shows the result of the question classification process. This question classification process is implemented with the help of Naïve Bayes Algorithm. The Questions observed in the output are extracted from tweets. Each question is categorized into any of the eight categories *like* Shoes, Model_merchendise, Games_Programme, Fan_question, Clothes, Video streaming, Enquiry_race, and Graphics_prints. E.g.: "When will Jenson Button Know about his #F1 Future?" This question clearly states the interest of a fan about his favorite F1 driver's future progress in the upcoming F1 races. That is the reason why the question has to fall under the "Fan_question" category.

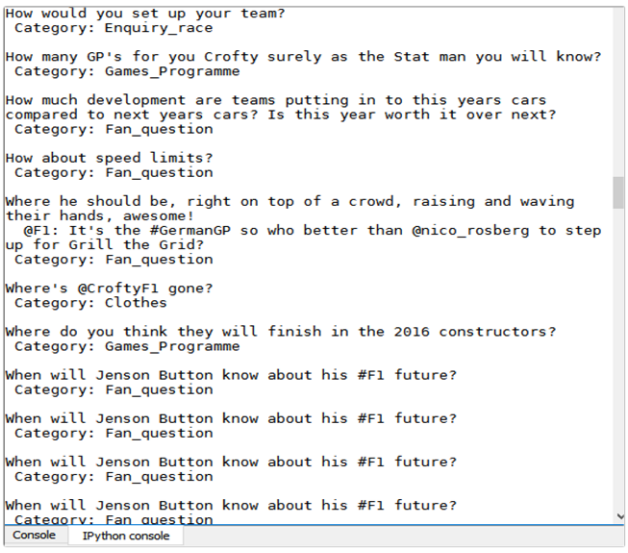


**Figure 5:** Output of the Classification

## REFERENCES


1. Pinto, D., M. Branstein, R. Coleman, M. King, W. Li, X. Wei, and W.B. Croft. Quasm: A system for question answering using semi-structured data. In Proceedings of the Joint Conference on Digital Libraries, 2002.
2. Radev, D. R., W. Fan, H. Qi, H. Wu, and A. Grewal. Probabilistic question answering from the web. In Proceedings of WWW-02, 11th International Conference on the World Wide Web.2002
3. Li, X. and D. Roth. Learning question classifiers. In Proceedings of the 19th International Conference on Computational Linguistics (COLING), 2002, pages 556–562.
4. Khardon, R. Machine Learning (1999) 35: 57. doi:10.1023/A:1007571119753
5. Hovy, E.H., U. Hermjakob, and D. Ravichandran. 2002a. A Question/Answer Typology with Surface Text Patterns. Proceedings of the Human Language Technology (HLT) conference. San Diego, CA.
6. Li, X. and D. Roth. 2002. Learning question classifiers. In Proceedings of the 19th International Conference on Computational Linguistics (COLING), pages 556–562.
7. Hacioglu, K. and W. Ward. Question classification with support vector machines and error correcting codes. In Proceedings of HLT-NAACL, 2003.
8. Zhang, D. and W. Lee. 2003. Question classification using support vector machines. In Proceedings of the 26th Annual International ACM SIGIR conference, pages 26–32.
9. Li, X. and D. Roth Learning Question Classifier the role of semantic Information. In Proceedings of Natural Language Engineering Journal Volume 12 Issue 3, September 2006 Pages 229 - 249
10. http://www.nltk.org/howto/wsd.html for Lesks Word Sense disambiguation Algorithm.
11. "Twitter4j" [Online]. Available: http://twitter4j.orgi
12. S. S. Asur and B. A. Huberman, “Predicting the future with social media,” in Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01, ser. WI-IAT ’10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 492–499. [Online]. Available: http://dx.doi.org/10.1109/WI-IAT.2010.63
13. Y. M. Li, H. M. Chen, J. H. Liou, and L. F. Lin, "Creating social intelligence for product portfolio design," Decls. Support Syst., vol. 66, 123-134, 2014.
14. D. Panicker, A. U, and S. Venkata krishnan, “Question classification using machine learning approaches”, International Journal of Computer Applications, 2012, 48(13):1–4.
15. O. Ferret, B. Grau, M. Hurault-Plantet, G. Illouz, L. Monceaux, R. Robba, and A. Vilnat, “Finding an answer based on the recognition of the question focus”, 2001, In TREC.
16. M. Heilman, “Automatic Factual Question Generation from Text”, PhD thesis. Carnegie Mellon University, USA, 2011.


**Source of Support:** Nil, **Conflict of Interest:** None.