Text-Based Intent Analysis using Deep Learning

4 authors, including:

Vraj Desai
VIT University
1 PUBLICATION   **0** CITATIONS

Some of the authors of this publication are also working on these related projects:

Project   Text-based Intent Analysis using Deep Learning View project

Text-Based Intent Analysis using Deep Learning

Vraj Desai1, Sidarth Wadhwa2, Anurag A3, Bhisham Bajaj4
1,2,3,4 School of Computing Science and Engineering, VIT University,
Vellore-632 014, India

Abstract:- Intent classification focuses on analyzing input to gauge the users' needs and opinions. With this proposed system, we utilise a neural network classifier that analyzes text strings that are fed to it as input. Based on the analyzed intent, the system responds to the user by acting upon the commands issued to it. The areas covered by the system include weather, ratings, and booking restaurants. Firstly the intent of the text will be analysed and then it will be classified based on the topic it belongs to. Due to expressiveness of intent by users intent , for instance "need a rental space", "places to visit in chennai" intent analysis plays a major role in e-commerce, real estate and online marketing industry. We aim to maximize the accuracy and minimize the errors and the time complextity of the proposed system.

Keywords:- Intent analysis, One Hot Encoding, Bag of words, Neural networks.

I. INTRODUCTION

In this age where people are always connected and always expressing themselves, it has become near impossible to gauge opinions about a subject, since there are so many of them being posted online for the world to see every second.

Intent classification is the process which involves association of text and finding out the specific purpose (intent) or a goal towards which the text is headed. This phenomenon is useful in understanding the intent of the users' comments, emails or chat conversations and provide results in the form of similarity in- dices which points to the similarity of the text towards a certain topic. The intent classifier is able to categorize text on the basis of intent in the same way how humans classify everything. The process basically works in two phases: first phase involves intent classification in which the text is classified into correspond- ing fields and second phase involves intent analysis in which the similarity of text towards the classified fields is calculated. The fields on which the intentions are divided involves opinion, news, complaint, suggestion, spam, marketing, etc. The applications of this phenomenon involve Spam detection, Advertising via social media, Customer Handling, etc. Analyzing what people are thinking and how they feel about certain products can give companies the insight and edge they need to beat out their competition by constantly improving their products. At the same time, any

automated assistance that aims to provide a multiplicity of service needs to be able to analyze a person's intent in order to judge what it is they require assistance with. Furthermore, review and opinion aggregator sites can utilize intent and/or sentiment analyses to get an idea about the general favorability or unfavourability of reviews being posted on their site.

These, among many other use cases, highlight the need for efficient intent anal- ysis. There is a need for accurate intent analysis systems that can judge what users require so that it can be served to them. Such systems have immense sig- nificance especially for the commercial sector, where their deployment can lead to streamlining and efficiency.

Neural networks aim to simulate the functioning of neurons inside the human brain. As such, if provided with enough of a dataset, they can be taught to pre- dict outputs across a wide ranging gamut of applications. This, combined with their high accuracy makes them a good choice for this system. Our proposed sys- tem aims to analyse text strings fed to it, and to provide the services required as stated therein. For instance, we aim to have the system use relevant websites to serve responses to weather queries. Basically, for this system we are going to analyse the intent of the text and then we will classify it accordingly as sunny, cloudy, rainy, mist and haze. For instance, we have a text "The temperature is 40 degree Celsius", this text will be analyzed and will be classified as "sunny". With this system, we aim to improve on the accuracy in analyzing intent and delivering relevant responses, of current solutions, we also aim at generating the results as fast as possible and hence reducing the time complexity of the system.s

II. LITERATURE SURVEY

Kroll et al [1] prepared a prototypical implementation of Intent Analysis for text documents. Provided a prototype of Intent Analysis that is a quick and proper visual evaluation of text in terms of Intent. The survey done on this prototype provided 57% accuracy.

Kaspars Balodis [2] prepared a system that detects the Intent of the user's ut- tered sentence or question. Provides training to word corpus in order to increase accuracy. Makes use of Bayesian approach instead of taking small data sets in order to provide greater accuracy. Automatic error-correction methods used in the system also improves the grammatical errors made by the user.

Ahmed Y. Tawfik [3] made use of cross-lingual query picking and result merging techniques to classify the intent of the user. Cross-Lingual Web Search(CLWS) introduced in this paper provides accuracy when it comes to poor languages like Arabic and even with more established languages like French.

Roberto González-Ibáñez [4] observed the contributions of linguistic and prag- matic features of tweets. Found out that 3 pragmatic features- ToUser, Smiley and Frown were the most discriminating features in this task. Best results achieved provided upto 71% accuracy. Also, this model provided feature ranking in order to measure the impact of features in falling into the 3 categories.

Madhumita Gupta [5] used three algorithms in this process of classifying the paragraphs into topics: Logistic Regression, Classification and Regression Trees, and Support Vector Machines. The pre-processing techniques like removal of stopwords and stemming enhances the probability of getting a higher accuracy towards a topic. Segregating paragraphs on the basis of different topics makes it more modular and easier to classify them rather than going for the Bag of Words and Topic Modeling approaches.

Yi Chang [6] incorporated click-approach on query documents in analyzing the intent. A model is prepared based on the ranking of such clicks for relevance improvement. Provides accuracy of intent of more than 80% and improves 18% accuracy for multi-regional intent detection and 15% accuracy for multilingual intent detection.

Xiao Ding [7] established a complete framework of CIMM in order to find out the consumption intention of the user. Neural networks like CNN has been used to train and build the model. The model can be used in psychological science in order to understand the patient needs. It is an effective method. Provides upto 71% accuracy.

As a survey, Arunima Jaiswal [8] used a descriptive and wide number of clas- sifying algorithms. Performance of RNNs are better. Also the shear number of algorithms show the varying performance for benchmarks.

S lawomir Dadas [9] measured the accuracy for original data, then the data is expanded for 50%, 100% and 200%. This provides a platform to randomly mu- tate the data, for better learning. The data expansion provides an overall better accuracy over the same model. The training is done with ADAM optimization algorithm for faster convergence.

Ming-Hsiang Su [10] created the architecture that uses multiple algorithms for sequential process of: Intent Detection, Evaluation of Dialog Policy, Sentence Generation and Appropriateness Test. A comprehensive output based dialog se- mantics over various algorithms. The word embedding alone is tested over 4 algorithms.

Lian Meng [11] used the hierarchical LSTM with an augmented memory module to utilize contextual information for dialog classification. The advantages over using an LSTM over an RNN for classifying long distance information. Accuracy is low for all used LSTMs. Classification accuracy however is slightly better.

Fantahun Bogale Gereme [12] did performance evaluations on multiple datasets using all mentioned methods. Works best with Deep CNNs. Accuracy has reached up to 99%. Works only on textual data.

To combat cyberbullying over social media, Sreeshma Mohan [13] suggests in- tent analysis for future warnings for negative and abusive texts. A combination of both cognitive psychology and deep learning is used here. A dataset was cre- ated ground up from scraping Github projects and tweets. Feature extraction and sentiment analysis was done for intent classification and identification – Hate, Offensive, etc.Accuracies are obtained for multiple ML algorithms, giving a wide range of results. The average accuracies are low.

Swati Agarwal [14] introduced a concept of semantic tagging which identifies role of each term in content which is an enhancement to keyword spotting methods. Experimental results generated clearly showed an edge over keyword spotting method. The comparison of the classification techniques i.e. Random forest tree, Naïve Bayes, decision tree etc. is based on only 2 datasets which is quite small. Thai-Le Luong [15] introduced LSTM-CRF models for intent analysis which are easy to work on in comparison to NLP. This is the only paper which focuses on intent extraction for Vietnamese text. Moreover all the technique used are experimentally verified. Only two domains are considered for experimental analysis.

Zhonghe Gao [16] used a concept of conceptual lattices which is semantic in nature and is used to look for relations between different contexts and analyze the text or data accordingly. Usage of concept-ual lattice allows to fetch more dense and richer intents as it is not only semantic preserving but also facilitates dimension addition.

Samantha Akulick [17] used N-gram based algorithm in the proposed system which divides the post into a list of n grams and checks them against predefined dictionaries which identify the intent associated with n-grams. Part of speech algorithm uses patterns which are unique combinations of POS tags and dic- tionaries. Third algorithm uses 30% of data for testing and 70% for building a model. A comparative analysis of all the three techniques used is generated which help us to understand that algorithm 1 is highly efficient (n-based) than the remaining two.

Thai-Le Luong [18] introduced BILSTM-BICRF models for intent analysis which are enhancement to the LSTM-CRF models proposed by the same author in 2017. These models provide tri-training which enhances the results accuracy. Tri-training is provided in the proposed

system which enhances the accuracy of the systems. Only three domains are considered for experimental analysis.

Amber Nigam [19] introduced a multi-staged approach in which problems are break down into stages and the result of one stage is passed to the other which reduces search complexity. Search space complexity is reduced for subsequent stages as the results at one stage are passed to the other.

In the system proposed by Cheyer et al [20], they obtain the user's request in the form of a text string. The request is received from the user in the form of speech input, and is then converted to a text string. A set of predefined domains is constructed and a plurality of words is associated with each. The text string obtained is then matched to these sets, in order to map the request to a partic- ular domain. The task is then performed in accordance with the predefined set of steps associated with the domain, in accordance with the parameters derived from the request.

Deshmukh et al [21] propose a system to detect the intent of a speaker by rank- ing words used in a conversation. To do this, they obtain a set of utterances in a conversation in chronological order as input and calculate the value of the 'intent confidence score' for each word. This metric is employed to help recognize the impact of a certain word on the subsequent conversation. To evaluate this, they use two criteria: Uniqueness of the word in the conversation and the frequency of utilization of the word in the ensuing exchange.

Purohit et al [22] suggest a system to analyze intent in short text social me- dia posts. Highlighting the ambiguity of interpretations and sparsity of relevant behaviors of such posts, they present a method for multiclass classification of intent. They utilize a hybrid feature set representation which combines both top down and bottom up approaches. For the former, they employ knowledge guided patterns, while using an n-gram bag-of-tokens model for the latter.

Park et al [23], in their paper, use status texts posted on social media as input to retrieve relevant apps. Addressing the problem by regarding the statuses as queries, and the mobile apps as the entities that need to be retrieved, they first build a parallel corpora from social media which contains implicit intention text as well as explicit intention text associated with the same. The user intentions are modelled using topic models, and the relevant apps are retrieved as per the predicted intention.

Cambria et al [24] highlight the significance of opinion mining and sentiment analysis in the online age. They express the fact that many companies nowadays con duct research on such techniques in order to get a better idea about the pulse of the market. They also highlight the need for these systems to exclude heated language in order to better filter relevant opinions.

Nesamoney et al [25] propose a system that utilizes brand awareness metrics such as Awareness, Consideration, Engagement, and Purchase Intent. They pro- pose interactive advertisements that can help gauge prospective buyers' intent, citing the opaqueness of metrics such as clicks.

III. RESEARCH GAP/FINDINGS

M. Kröll [1] introduces a new method/prototype for analyzing goals and inten- tions of text found in various web documents by making use 44 transcripts of political speeches from American Presidential Candidates from 2008. Ahmed Y. et al [3] examined query intent classification and improved the performances of cross-lingual information retrieval (CLIR). A. Jaiswal [8] made use of 4000 tweets to identify the humor involved on those tweets. But the main problem with these approaches was that the data sets used were very limited which led to a decrease in accuracy. R. González-Ibáñez [4] applied machine learning tech- niques on Twitter comments in order to detect the sarcasm hidden in those comments. But the length of tweets makes it difficult to classify them into cate- gories and detecting sarcasm for machines is difficult just like it is with humans.

Y. Chang [6] performed Intent Analysis on documents by taking clicks as in- puts from the users. An n-gram language models(LM) are built on the basis of such query click probabilities and these LMs are trained using data sets. But web search ranking is not inculcated in this model and also the model does not consider the session level clicks. M. Gupta [5] took inputs as word documents and then Then each paragraph is given a topic to which it belongs after pre- processing the text (removing stopwords, stemming, Lower-casing the text, etc.).

Then the Document Term Matrix is taken as input and various machine learning algorithms are implemented on it in order to classify it.division of the document into paragraphs is conducted. But building a taxonomy with thousands of words becomes a very difficult and expensive task. Also, it does not show much dif- ference when it comes to more general topics and more specific topics S. Dadas [9] used a compound neural architecture which uses a layered approach on the ATIS dataset. But in this approach, accuracy decreases because of duplication in sentences which is caused by high probabilities of words. S. Akulick [17] aimed to identify customer's intents using three techniques: analysis using n-grams, analysis using part of speech tagging and analysis by combining n-grams and support vector machines by making use of dictionaries. But dictionaries are dif- ficult to maintain and are often restricted to single language. Thai-Le Luong [15] focused on extracting social media texts in an enhanced way by constructing an ensemble neural network, to achieve this two machine learning techniques are used namely BICRF and BILSTM. But usage of such techniques led to an in- crease in time complexity.Park et al, in their paper, use status texts posted on social media as input to retrieve relevant apps. But making use of social media posts can often be hard to comprehend and might not always provide the best estimation of what a user needs.

With our proposed system, the aim is to develop a system utilizing Natural Lan- guage Toolkit and Neural Networks that can analyze a user's intent based on the input text and serve relevant services. We also aim to increase accuracy and reduce time complexity of the model.

IV. METHODOLOGY

Classifying large number of queries requires Artificial Neural Networks (ANN). The generic process is acquiring the raw data, preprocessing to clean the data and iterating it through the Artificial Neural Network (code, test and tune the model). An Artificial Neural Network involves a series of algorithms that are used to recognize patterns in the data in the same way as the human brain. The data acquired is preprocessed initially to clean the data and remove all the unwanted characters/words in order to increase the accuracy of the model. This cleaning phase involves stemming the sentences, removing the stopwords and removing any unwanted characters from the data. Then the data is then divided into Bag of Words and then the data is analyzed using machine learning and finally the data is put into one of the categories in accordance to its intent.
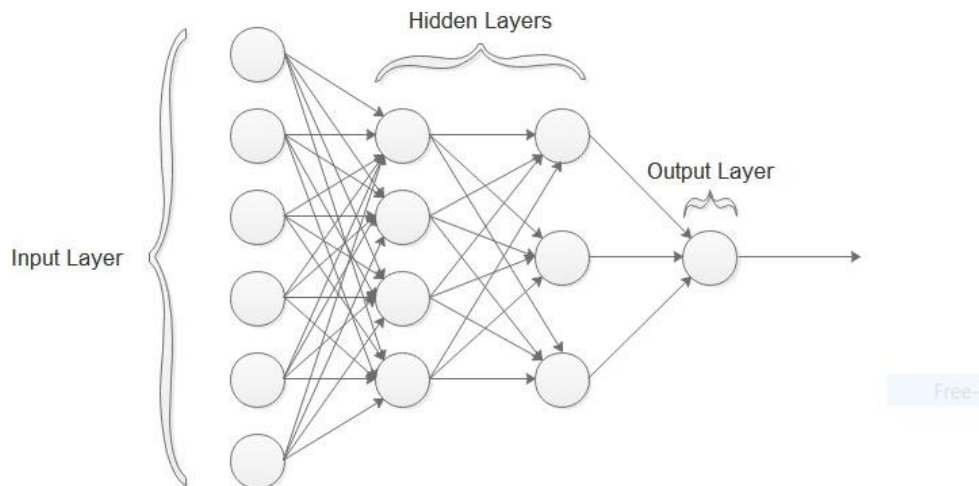

Fig 1:- Artificial Neural Network

➤ *Acquiring the data*
The training data is to be acquired from a laptop for Google Console Search data. This provides the generic search data a regular person actually looks for. Getting such data proves that the data is clean and tampered with.

➤ *Preprocessing the data*
This data is often not "clean". Preprocessing is done so as to remove unwanted characters, grouping similar data and in general making it more legible for the machine. Unwanted characters generally occur due to bad read, creating an in- significant loss of data when removing them. This can be done by running a regular expression through the data as it is being read. Then, the data is to be lemmatized. Lemmatizing refers to grouping various inflected forms of the word so that they can collectively be referred by a single entity. For example: "days" -> "day". After stemming the words, the data is reduced to 0s and 1s matching only the unique words in the corpus.

• *Bag of words:*
Say the bag contains the following after stemming: ["there","you","are","a"] Input Sentence: "a car". The following will generate bag of words: [0,0,0,1]. This matches only the index of "a" since car doesn't exist in the bag. Similarly output will also be in the form of a bag, if there are 3 intents, say location, thing and feeling, generated output will be [0,1,0].

➤ *Neural Networks*
After cleaning the data, a sigmoid activation function is used to normalize and adjust the error rate. A sigmoid function is an S shaped curve ranging from 0-1, therefore generally used in models where probability is the final output. A sigmoid is differentiable, meaning slope can be found given 2 points, so even it's derivative can be used in lowering error rates. A bag of words function is created for transforming the input statement to 0s and 1s. This coincides with the transform applied to the training data.

A two-layered Neural Network is to be used. This translates to 1 input layer, 2 hidden layers and 1 output layer. The number of neurons used in the hidden layers can be adjusted to get low error rates.

The function for training the neural network, adds synaptic weights to each neuron. This refers to the amplitude of a connection between two nodes. An alpha gradient descent parameter can be used to fine tune the error rate. It is a first-order iterative optimization algorithm and it's goal is to find the minimum loss in the process of training the model with different weights or updating weights.
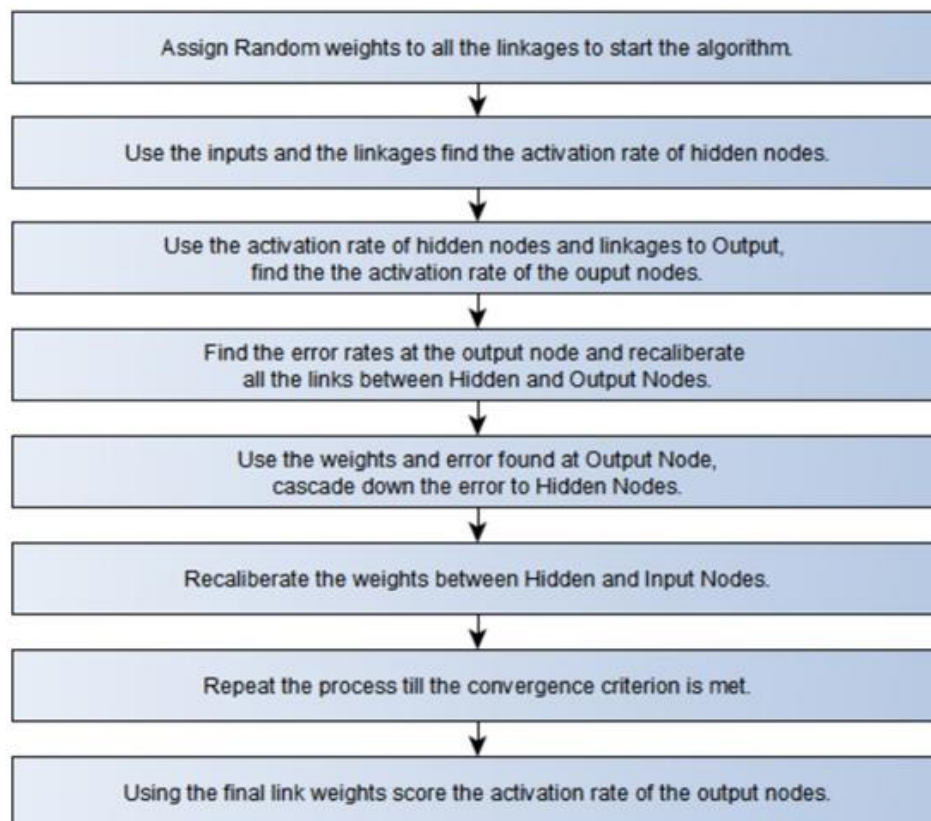
Fig. 2: Iterating Procedure of the proposed Neural Network

➢ *Text Encoding*

To analyze the given text we have to assign integer variables to the tokens or words, on getting those numbers (integer variables) machine learning algorithms are used to cluster and classify the data based on data analytics. Text Encoding can be classified into four techniques:

Document Vectorization This encoding technique is also known as one hot(non- ordered) encoding. The model is based on dictionary as every word present in the document is assigned with value 1 and hence we obtain the document vec- tor consisting of 0's and 1's. The main disadvantage of this technique is that it doesn't preserve the order of words which is important in case of negation sentence.

One hot Encoding (Ordered) This model uses Recurrent Neural networks along with LSTM to preserve the sequence of the words which leads to better classification, it overcomes the disadvantage of Document vectorization i.e. or- der of words is preserved. The document consists of large sequence of 0's and 1's tensors.

Index based Encoding This model uses a dictionary to map each word to the index thereby preserving the order of words. A numerical distance between words is introduced between the words.

Word Embedding In this technique a numerical vector is associated between words in the dictionary and the semantic meaning is captured between words at different numeric distance. In this way a geometrical space is formed by the vectors which is known as embedding space.

Word2Vec and Glove are the two most prominent embedding techniques used for text processing and classification.

➢ *Bag of Words*

In this technique we focus on words which are referred as tokens and are atomic units for the text processing and analyzing the intent of the text. There is no need to search for each word in the given text, we only have to focus on contribut- ing words i.e. content providing and meaningful words. The other non-content providing words are referred as stop words. For example in the sentence The largest Ancient wall word 'the' has no significance and hence it can be referred as a stop word. This technique is highly efficient when all the word in a text differ in weights.

V. ALGORITHM

For our dataset, we create our own sentences with their associated classes, in order to test the algorithm with clean, self-generated data. External datasets on this subject often requires extensive cleaning. We utilise the natural language toolkit (nltk) for tokeization, and to split the data into documents, classes and words. Additionally, the words are 'stemmed', which allows the algorithm to recognize words with different suffixes, but with the same stems. As a result of this, 'sandwich' and 'sandwiches' will both be recognized as the same, with the case causing no exception. The data is then transformed into a bag-of-words model, and the training data is matched against the unique words in this bag, reducing the computational complexity as most of the data is dealt with in terms of 0s and 1s.

The data is then presented to a two-layer neural network for training. After training and fine tuning the learning rate for gradient descent to obtain optimal convergence, the model is saved as a JSON format file.

The number of neurons are also fine tuned, starting from the base number of 20, to obtain accurate classification. In case multiple predictions are generated, a threshold value is set for the confidence metric, in order to filter out the less likely prediction.
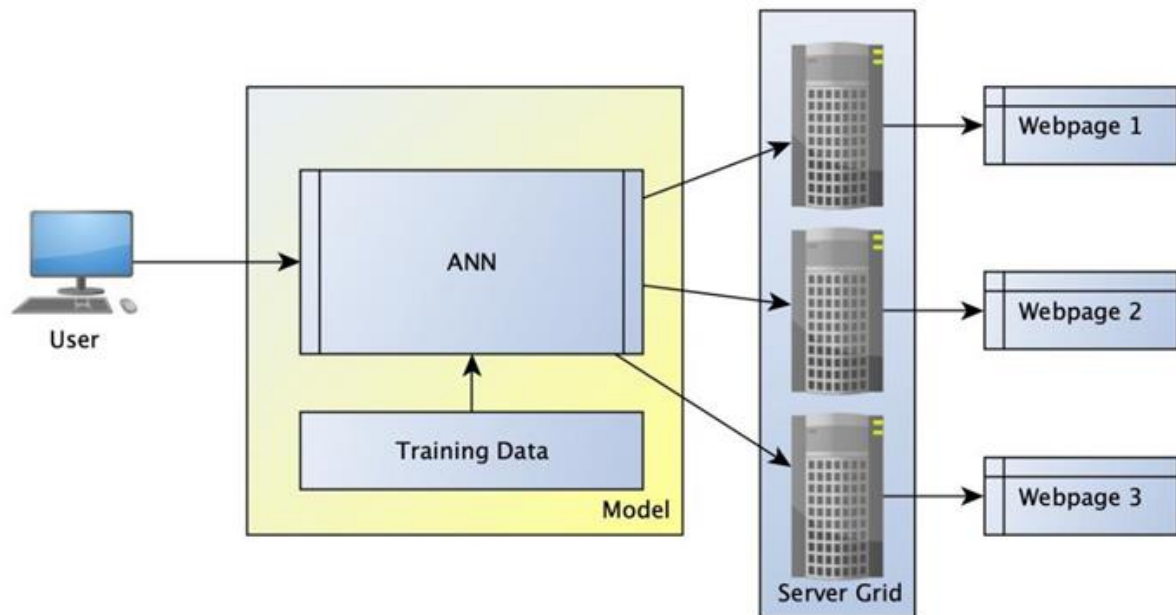
$$\_q\,0+ = \alpha\boxed{}q\_0\ \ weight\ update$$

Fig 3:- Request to Response Model for our use case

VI. TRAINING DATA

Training data is generated by making a dictionary based approach for class and a sample sentence. A key value pair is made for each of the aforementioned cat- egories. For example, the "class" key is mapping to a categorical value, similarly the "sentence" key maps to a sample request question. The dataset being used is created by us to generate a cleaner version. There are intentional differences in spelling for the same word. This is done to test the efficiency of bag of words functionality.

The key value pairs generated are appended to a list, which consolidates all the data. The list is sent as a parameter to the classify function, which in turn tok- enizes the sentences. This is stored in the variable "words", which is generated by the stemmer.

VII. VALIDATION

The training time varies for different sizes of data. The input matrix for the classification has the dimensions:

$$(Num\ of\ Rows)\boxed{}(Total\ Stemmed\ Word)$$

For this particular run of the algorithm we get 53 unique stemmed words and 28 rows which is the initial data. A composite value of time elapsed cannot be calculated based on the request. The elapsed time is calculated for the given request by finding the difference between the start and end times for training. For a 28 row

data alone the training time is about 8.5 seconds. A synapse file is generated to store all recorded data in JSON format.

The output matrix has the dimensions:

$$1\boxed{}(Total\ Fields\ Classified)$$

A request, "What matches are on today?" is used as a validator. This is taken as a user input and the classify function generates the results based on the given request. After it "thinks" where to place it, the output matrix is simply put through an if-else ladder. A pre-existing database of trusted websites over a switch case is used as a response model. For an example query, "Make some pasta" gives an 85% accuracy to "food" classification. This is switched to the case which holds the URL for "food" section and redirects the user to this URL.

VIII. COMPARATIVE ANALYSIS

After training the model we tested some queries and verified them with the re- quest to response manually. So analysing from table 1, we conclude by adding up the correct and partially correct matches by saying the average accuracy is 77.6%. This is done on a relatively low number of tests as human verification is a must.

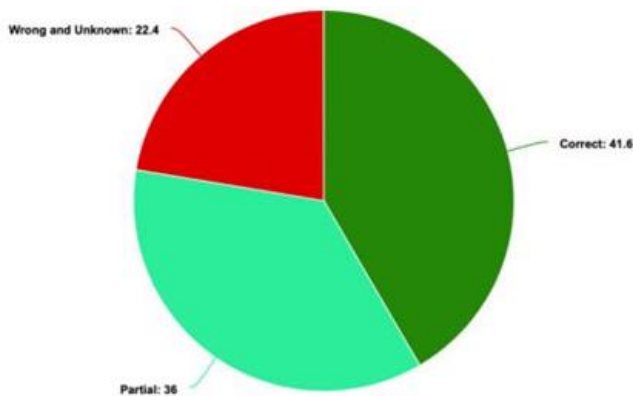| Type of Result	Percentage of Total Test Queries
Correct Match	41.6%
Partial Correct Match	36%
Wrong and Unknown	22.4%

Table 1:- Test Data Analysis

Fig 4:- Test Data Segregation

We outcasted M Kroll's [1] model in terms of accuracy, the system proposed by the former and the prototypic implementation for the same resulted in a 57% accuracy whereas our proposed model using deep learning is more efficient and reliable with 77.6% accuracy. M. Gupta [5] the algorithm proposed by the au- thor includes pre-processing techniques like removal of stop words and stemming which enhances the probability of getting a higher accuracy towards a topic. In our model we have combined the pre-processing techniques with a topic mod-elling Approach "Bag of words" to increase the accuracy of our system to some more extent. L. Meng [11] made use of LSTM along with CNN leading to a Deep CNN model which definitely had advantage over our model leading to 99% accu- racy, our model has 77.6% accuracy but for this little advancement, the overall complexity of the system became highly complex. Moreover, it was restricted to two topics(model proposed by Meng) whereas our model covered more than 10 topics leading to the larger dataset over the former. S. Mohan [13] implemented the intent analysis in detecting abusive content in social media platforms taking cognitive psychology into the account in place of pre-processing techniques but the accuracy of the model proposed by the author was low (63%).

IX. CONCLUSION

Model	Accuracy
LSTM	74.5%
HLSTM	76.3%
HLSTM+Mem	76.7%
ANN (Our Model)	77.6%

Table 2:- Comparison of models [11]

This paper implements a Artificial Neural Network (ANN) architecture for clas- sifying text based queries which generates a response redirect website. Through tests and trials, we conclude that this method performs slightly better than pre- vious approaches. The pre-processing used gives this model the advantage as it mainly searches for words of interest and classifies similar words as a unique word.

Future research includes expanding the dataset to handle errors from new words, emoji and other lingo. This fairly increases the application prospects to target a bigger audience. Increasing dataset results in better classifying unknown classes and the model understands the intent more like a human.

REFERENCES

[1]. M. Kröll and M. Strohmaier, "Analyzing human intentions in natural language text," in *Proceedings of the fifth international conference on Knowledge capture*, pp. 197–198, 2009.

[2]. K. Balodis and D. Deksne, "Fasttext-based intent detection for inflected lan- guages," *Information*, vol. 10, no. 5, p. 161, 2019.

[3]. A. Y. Tawfik and A. S. Kamel, "Query language and query intent in cross-lingual web search,"

[4]. R. Gonzalez-Ibanez, S. Muresan, and N. Wacholder, "Identifying sarcasm in twit- ter: a closer look," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers-Volume 2*, pp. 581–586, Association for Computational Linguistics, 2011.

[5]. M. Gupta and S. Guha, "Topic based analysis of text corpora," *Computer Science & Information Technology*, p. 33, 2016.

[6]. Y. Chang, R. Zhang, S. Reddy, and Y. Liu, "Detecting multilingual and multi- regional query intent in web search," in *Twenty-Fifth AAAI Conference on Artifi- cial Intelligence*, 2011.

[7]. X. Ding, T. Liu, J. Duan, and J.-Y. Nie, "Mining user consumption intention from social media using domain adaptive convolutional neural network," in *Twenty- Ninth AAAI Conference on Artificial Intelligence*, 2015.

[8]. A. Jaiswal, A. Mathur, S. Mattu, *et al.*, "Automatic humour detection in tweets using soft computing paradigms," in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pp. 172–176, IEEE, 2019.

[9]. S. Dadas, J. Protasiewicz, and W. Pedrycz, "A deep learning model with data enrichment for intent detection and slot filling," in *2019 IEEE International Con- ference on Systems, Man and Cybernetics (SMC)*, pp. 3012–3018, IEEE, 2019.

[10].M.-H. Su, C.-H. Wu, and L.-Y. Chen, "Attention-based response generation using parallel double q-learning for dialog policy decision in a conversational system," *IEEE/ACM*

Transactions on Audio, Speech, and Language Processing, vol. 28, pp. 131–143, 2019.

[11]. L. Meng and M. Huang, "Dialogue intent classification with long short-term memory networks," in *National CCF Conference on Natural Language Processing and Chinese Computing*, pp. 42–50, Springer, 2017.

[12]. F. B. Gereme and W. Zhu, "Early detection of fake news" before it flies high"," in *Proceedings of the 2nd International Conference on Big Data Technologies*, pp. 142– 148, 2019.

[13]. S. Mohan, I. Valsaladevi, and S. M. Thampi, ""think before you post": A cognitive psychological approach for limiting bullying in social media," in *International Conference on Smart City and Informatization*, pp. 418–434, Springer, 2019.

[14]. S. Agarwal and A. Sureka, "But i did not mean it!—intent classification of racist posts on tumblr," in *2016 European Intelligence and Security Informatics Confer- ence (EISIC)*, pp. 124–127, IEEE, 2016.

[15]. T.-L. Luong, M.-S. Cao, D.-T. Le, and X.-H. Phan, "Intent extraction from social media texts using sequential segmentation and deep learning models," in *2017 9th International Conference on Knowledge and Systems Engineering (KSE)*, pp. 215–220, IEEE, 2017.

[16]. Z. Gao, Y. Nie, and Y. Wang, "An approach to text knowledge analysis based on many-valued context," in *2009 International Conference on Management and Service Science*, pp. 1–4, IEEE, 2009.

[17]. S. Akulick, E. S. Mahmoud, *et al.*, "Intent detection through text mining and analysis," in *Future Technologies Conference (FTC)*, 2017.

[18]. T.-L. Luong, N.-T. Tran, and X.-H. Phan, "Improving intent extraction using ensemble neural network," in *2019 19th International Symposium on Communica- tions and Information Technologies (ISCIT)*, pp. 58–63, IEEE, 2019.

[19]. A. Nigam, P. Sahare, and K. Pandya, "Intent detection and slots prompt in a closed-domain chatbot," in *2019 IEEE 13th International Conference on Semantic Computing (ICSC)*, pp. 340–343, IEEE, 2019.

[20]. A. J. Cheyer, C. D. Brigham, and D. R. Guzzoni, "Determining user intent based on ontologies of domains," Jan. 27 2015. US Patent 8,942,986.

[21]. O. D. Deshmukh, S. Joshi, S. Saurabh, and A. Verma, "Intent discovery in audio or text-based conversation," Mar. 17 2015. US Patent 8,983,840.

[22]. H. Purohit, G. Dong, V. Shalin, K. Thirunarayan, and A. Sheth, "Intent classification of short-text on social media," in *2015 ieee international conference on smart city/socialcom/sustaincom (smartcity)*, pp. 222–228, IEEE, 2015.

[23]. D. H. Park, Y. Fang, M. Liu, and C. Zhai, "Mobile app retrieval for social media users via inference of implicit intent in social media text," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pp. 959–968, 2016.

[24]. E. Cambria, B. Schuller, Y. Xia, and C. Havasi, "New avenues in opinion mining and sentiment analysis," *IEEE Intelligent systems*, vol. 28, no. 2, pp. 15–21, 2013.

[25]. D. Nesamoney, P. S. Chandra, D. Charak, S. Dahiya, S. Kumar, and H. Guntren, "User intent analysis engine," May 31 2012. US Patent App. 13/297,064.