Laptop Recommendation Chat Bot

Anshuman Singh, Yashwanth Reddy
School of Computer Science and Engineering
Vellore Institute Of Technology, Chennai.

Abstract

Natural Language Processing (NLP) is basically how you can teach machines to understand human languages and extract meaning from text. Language as a structured medium of communication is what separates us human beings from animals. We are surrounded by text data all the time sourced from books, emails, blogs, social media posts, news and more. Natural Language Processing is expected to be worth 30 Billion USD by 2024 with the past few years seeing immense improvements in terms of how well it is solving industry problems at scale. Natural language processing includes many methodologies that have the capability of understanding and producing natural language as used by human beings regardless of that language is. We will be using some methodology to try and understand human language and create a chat bot that can converse with humans over the internet and make useful recommendations about laptops to the users An NLP based chatbot is a computer program or artificial intelligence that communicates with a customer via textual or sound methods. Chatbots are applications that imitate human conversations for solving various tasks. Everything we express in written or verbal form encompasses a huge amount of information that goes way beyond the meaning of individual words. The combination of topic, tone, selection of words, sentence structure, punctuation/expressions allows humans to interpret that information, its value, and intent. Theoretically, humans are programmed to understand and often even predict other people's behavior using that complex set of information.

1 Introduction

In this project we will try to create a chat bot that will have only one purpose which will be to recommend laptops to the user based on their question. The idea is that the users will open start talking to the chat bot and they will ask a question, based on that question the chat bot will generate a fitting response and send it to the user. We are using two different approaches for this purpose and will be comparing their performance. The first approach is to use simple Tf-Idf approach to generate responses. The second approach is to use deep learning and bag or words approach. As the name suggests, bag of word, the concept is to create a bag of words from the clutter of words, which is also called as the corpus. It is the simplest form of representing words in the form of numbers. We convert the words to digits because the system needs the information in the form of numbers, or else it won't be able to process the data.We convert the words to numbers by analyzing the presence of the word in a particular sentence. A number is denoted as an encoded value against the word. This is the number of times that word has been represented in the sentence. If only the presence is to be considered, then the game is denoted in form 1's and 0's. When the word is present in the sentence, it is denoted as 1 else 0. This is called a binary bag of words. Tf-idf is a method which is based on the frequency method but it is different to the bag-of words approach in the sense that it takes into account not just the occurrence of a word in a single document but in the entire corpus. TF-IDF works by penalising the common words by assigning them lower weights while giving importance to words which are rare in the entire corpus but appear in good numbers in few documents. Important terms related to TF-IDF:
TF (Number of times term t appears in a document)/(Number of terms in the document)
IDF=log(N/n), where, N is the number of documents and n is the number of documents a term t has appeared in.
TF-IDF=TF'IDF

2 Related Work

This section briefy reviews the literature, Based on our research, no chatbots exist for the sole purpose of recommending laptops to users. In our research we read few research papers the literature review is given below.

2.1) Twitter Bots and Gender Detection using Tf-idf Notebook for PAN at CLEF 2019, As the amount of unstructured data increases, value (and the number) of models that can infer information from this data also increases. This paper presents another such model that can perform bots and gender detection on Twitter using just the tweets from the respective Twitter user. We show that a simple frequency based approach with a machine learning algorithm i.e., SVM can achieve high accuracy if the preprocessing is done right. In English language. our model detects bots with an accuracy of 91 and gender with an accuracy of 82. Main strength of this model is its simplicity along-with the ease with which it can be used with other languages.

2.2) Author Profiling: Bot and Gender Prediction using a Multi-Aspect Ensemble Approach Notebook for PAN at CLEF 2019. Author Profiling is one of the most important tasks in authorship analysis. In PAN 2019 shared tasks, the gender identification of the author is the main focus. Compared to the previous year the author profiling task is expended by having documents written by bots. In order to tackle this new challenge we propose a two phase approach. In the first phase we exploit the TF-IDF features of the documents to train a model that learns to detect documents generated by bots. Next, we train three models on character-level and word-level representations of the documents and aggregate their results using majority voting. Finally, we empirically show the effectiveness of our proposed approach on the PAN 2019 development dataset for author profiling.

2.3) Bots and Gender Profiling Using a Deep Learning Approach Notebook for PAN at CLEF 2019.This paper describes the system we developed for the Bots and gender profiling task, at PAN @ CLEF 2019. The task consists in, given a tweets set, automatically determine whether its author is a bot or a human. In case of human, identify her/his gender. We propose a deep learning based system, fed with the TF-IDF representation from the texts instead of word embeddings representation as usual. Additionally, we use some linguistic features which improve the performance of the system according with the experimental results.

2.4) An Improved Text Sentiment Classification Model Using TFIDF and Next Word Negation.With the rapid growth of Text sentiment analysis, the demand for automatic classification of electronic documents has increased by leaps and bound. The paradigm of text classification or text mining has been the subject of many research works in recent time. In this paper we propose a technique for text sentiment classification using term frequency- inverse document frequency (TF-IDF) along with Next Word Negation (NWN). We have also compared the performances of binary bag of words model, TF-IDF model and TF-IDF with 'next word negation' (TF-IDF-NWN) model for text classification. Our proposed model is then applied on three different text mining algorithms and we found the Linear Support vector machine (LSVM) is the most appropriate to work with our proposed model. The achieved results show significant increase in accuracy compared to earlier methods.

2.5) Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents. In this paper, the use of TF-IDF stands for (term frequency inverse document frequency) is discussed in examining the relevance of key-words to documents in corpus. The study is focused on how the algorithm can be applied on number of documents. First, the working principle and steps which should be followed for implementation of TF-IDF are elaborated. Secondly, in order to verify the findings from executing the algorithm, results are presented, then strengths and weaknesses of TD-IDF algorithm are compared. This paper also talked about how such weaknesses can be tackled. Finally, the work is summarized and the future research directions are discussed.

2.6) Hot Topic Detection Based on a Refined TF-IDF Algorithm. In this paper, we propose a refined term frequency inversed document frequency (TF-IDF) algorithm called TA TF-IDF

to find hot terms, based on time distribution information and user attention. We also put forward a method to generate new terms and combined terms, which are split by the Chinese word segmentation algorithm. Then, we extract hot news according to the hot terms, grouping them into K-means clusters so as to realize the detection of hot topics in news. The experimental results indicated that our method based on the refined TF-IDF algorithm can find hot topics effectively.

2.7) Detecting Hate Speech and Offensive Language on Twitter using Machine Learning: An N-gram and TFIDF based Approach. Toxic online content has become a major issue in today's world due to an exponential increase in the use of internet by people of different cultures and educational background. Differentiating hate speech and offensive language is a key challenge in automatic detection of toxic text content. In this paper, we propose an approach to automatically classify tweets on Twitter into three classes: hateful, offensive and clean. Using Twitter dataset, we perform experiments considering ngrams as features and passing their term frequency-inverse document frequency (TFIDF) values to multiple machine learning models. We perform comparative analysis of the models considering several values of n in n-grams and TFIDF normalization methods. After tuning the model giving the best results, we achieve 95.6 accuracy upon evaluating it on test data. We also create a module which serves as an intermediate between user and Twitter.

2.8) Unsupervised Sentence Representations as Word Information Series: Revisiting TF–IDF. Sentence representation at the semantic level is a challenging task for Natural Language Processing and Artificial Intelligence. Despite the advances in word embeddings (i.e. word vector representations), capturing sentence meaning is an open question due to complexities of semantic interactions among words. In this paper, we present an embedding method, which is aimed at learning unsupervised sentence representations from unlabeled text. We propose an unsupervised method that models a sentence as a weighted series of word embeddings. The weights of the word embeddings are fitted by using Shannon's word entropies provided by the Term Frequency–Inverse Document Frequency (TF–IDF) transform. The hyperparameters of the model can be selected according to the properties of data (e.g. sentence length and textual gender). Hyperparameter selection involves word embedding methods and dimensionalities, as well as weighting schemata. Our method offers advantages over existing methods: identifiable modules, short-term training, online inference of (unseen) sentence representations, as well as independence from domain, external knowledge and language resources. Results showed that our model outperformed the state of the art in well-known Semantic Textual Similarity (STS) benchmarks. Moreover, our model reached state-of-the-art performance when compared to supervised and knowledgebased STS systems.

2.9) Research on Text Classification Based on Improved TF-IDF Algorithm.In solving the problem of feature weight calculation for automatic text classification, we use the most widely used TF-IDF algorithm. Although the algorithm is widely used, there is a problem that the feature categories have different weights when calculating the weights. This paper proposes an improved TF-IDF algorithm (TFIDCRF) that takes into account the relationships between classes to complete the classification of texts. By modifying the calculation formulas of IDF to correct the problem of insufficient classification of feature categories, the naive Bayes classification algorithm is used to complete the classification. Finally, the proposed algorithm is compared with two others improved TFIDF algorithms. The results of the three text classification evaluation indicators show that the proposed algorithm has certain advantages in text classification.

3 Proposed Work

We will be using two methods in order to create this chat bot.
3.1) The first approach will be using the TF-IDF approach and the second will be to use deep learning using bag of words and Tf learn library in python. Tf- IDF stands for Term frequency – Inverse Document Frequency ,it is a technique to quantify words in a set of documents. We generally compute a score for each word to signify its importance in the document and corpus.

tf-idf stands for term freq - inv document

freq, it's a simple method to try to give scores to documents that look the same as the user's query. TF-IDF can be broken down into two parts TF (term frequency) and IDF (inverse document frequency).
What is TF (term frequency)?

Term frequency works by looking at the frequency of a particular term you are concerned with relative to the document. There are multiple measures, or ways, of defining frequency:
a. Number of times the word appears in a document (raw count).
b. Term frequency adjusted for the length of the document (raw count of occurences divided by number of words in the document).
c. Logarithmically scaled frequency (e.g. log(1 + raw count)).
d. Boolean frequency (e.g. 1 if the term occurs, or 0 if the term does not occur, in the document).

What is IDF (inverse document frequency)?
Inverse document frequency looks at how common (or uncommon) a word is amongst the corpus. IDF is calculated as follows where t is the term (word) we are looking to measure the commonness of and N is the number of documents (d) in the corpus (D).. The denominator is simply the number of documents in which the term, t, appears in.

Data Needed to Calculate TF-IDF

To calculate Term Frequency - Inverse Document Frequency (TF-IDF) the importance of the words in a document are scored on how frequently they occur across multiple documents. Words that are very common like "the" or "a" will be scaled down while words that appear often in a single document will be scaled up. The term frequency value should be normalized to take different sized documents into account, and very high term frequency values should be regarded as suspicious.

Let t=Term
Let IDF=Inverse Document Frequency
Let TF=Term Frequency

$$\text{TF} = \frac{termfrequencyindocument}{totalwordsindocument}$$

$$IDF(t) = \log_2\left(\frac{totaldocumentsincorpus}{documentswithterm}\right)$$

Calculate TF
As stated earlier:
$$\text{TF} = \frac{termfrequencyindocument}{totalwordsindocument}$$

Calculate IDF
Again the IDF equation is:
$$\text{IDF(t)} = \log_2\left(\frac{totaldocumentsincorpus}{documentswithterm}\right)$$

Now, In the TF-Idf approach for making our chatbot what we have done is:

- Importing necessary Libraries like nltk, pandas, TfidfVectorizer, and cosinesimilarity.
- Upload Self-Made Dataset
- After that remove special characters and empty spaces
- In the next stage we will do Preprocessing and Cleaning Data
- Remove punctutation
- Tokenize input
- Lmmatize words
- Setting up the bot add greeting inputs and responses
- return a random greeting from a pre-defined list
- In the next stage will start Talking to the bot
- Adding user input to list of tokens for comparison
- convert tokens into a vector
- find cosine similarity of user input (last item in list) with article vectors
- output corresponding answer from answer column

Limitations of Bag-of-Words The bag-of-words model is very simple to understand and implement and offers a lot of flexibility for customization on your specific text data.

It has been used with great success on prediction problems like language modeling and documentation classification.

Nevertheless, it suffers from some shortcomings, such as:

Vocabulary: The vocabulary requires careful design, most specifically in order to manage the size, which impacts the sparsity of the document representations. Sparsity: Sparse representations are harder to model both for computational reasons (space and time complexity) and also for information reasons, where the challenge is for the models to harness so little information in such a large representational space. Meaning: Discarding word order ignores the context, and in turn meaning of words in the document (semantics). Context and meaning can offer a lot to the model, that if modeled could tell the difference between the same words differently arranged ("this is interesting" vs "is this interesting"), synonyms ("old bike" vs "used bike"), and much more.

3.2) Our next approach is Bag of words which is a deep learning approach in this approach The bag-of-words model is used to preprocess the text by converting it into a bag of words or fixed-length vectors, machine learning algorithms. It is the simplest form of text representation in numbers. It is extremely easy, both to understand and to implement, and is used for language modeling and document classification.

It is a way to extract features from text to be used in modeling.

A bag-of-words includes a vocabulary of known words and a measure of the presence of known words. It describes the occurrence of words in a document.

The model only bothers only about whether known words show up in the document. It does not care where they show up in the document, only that they do show up.

It tries to learn about the meaning of a document from its content alone and assumes that if documents have similar content, they are similar to each other.

We cannot directly feed text into algorithms applied in NLP. They work on numbers. The model converts the text into a bag-of-words. The bag-of-words keeps a count of the occurrences of the most frequently occurring words in that text.

The model counts the number of times each word appears and turns text into fixed-length vectors.

How does the bag-of-words model work?
Here are the steps involved in implementing the bag-of-words model:

The first step is to pre-process the data. The text needs to be converted into lower case, all non-word characters need to be removed, and all punctuations need to be removed.

In the next step, we have to find the most frequent words in the text. The vocabulary must be defined, each sentence must be tokenized to words, and then the number of times the word occurs must be counted.

After that, the model is constructed. A vector is built to determine whether a word is a frequent word. If it is a frequent word, it is set as 1 and if not, it is set as 0.

And now you get your output.

In Bag of words approach what we have done is:

- Importing necessary Libraries like nltk, pandas, TfidfVectorizer, and cosinesimilarity.
- Upload Self-Made Dataset
- Removing any duplicates that have already been seen.
- using one hot encoding for scoring The simplest scoring method is to mark the presence of words as a boolean value, 0 for absent, 1 for present.
- look through the labels list, see where a tag is, set that value to 1 in the output row
- Training model, turn into np array to work woth tflearn model
- Creating Model
- make sure to get rid of all previous settings (if any)
- Now that we have preprocessed all of our data we are ready to start creating and training a model. For our purposes we will use a fairly standard feed-forward neural network each layer will have 8 neurons
- Making Predictions Talking to the bot
- This makes the process to generate a response look like the following:
  – Get some input from the user

– Convert it to a bag of words
– Get a prediction from the model
– Find the most probable class
– Pick a response from that class

What is the biggest advantage of the bag-of-words model?
The most significant advantage of the bag-of-words model is its simplicity and ease of use. It can be used to create an initial draft model before proceeding to more sophisticated word embeddings.
What are the limitations and disadvantages of the bag-of-words model?
The bag-of-words model is rather easy to understand and implement, but it does have some limitations and drawbacks. The vocabulary/dictionary needs to be designed very carefully. Its size has an impact on the sparsity of the document representations and must be managed well. The model ignores context by discarding the meaning of the words and focusing on the frequency of occurrence. This can be a major problem because the arrangement of the words in a sentence can completely change the meaning of the sentence and the model cannot account for this. Another major drawback of this model is that it is rather difficult to model sparse representations. This is due to informational reasons as well as computational reasons. The model finds it difficult to harness a small amount of information in a vast representational space.
Difference between bag-of-words and TF-IDF?
Term frequency-inverse document frequency (TF-IDF) is a numerical statistic the purpose of which is to reflect how important a word is to a document in a collection or corpus." Term Frequency (TF) is a measure of how frequently a term, t, appears in a document. Inverse Document Frequency (IDF) is a measure of how important a term is. The IDF value is important because simply computing just the TF is not sufficient enough to understand the importance of words. The main differences between bag-of-words and TF-IDF are that: Bag of Words only creates a set of vectors that contains the count of word occurrences in the document (reviews). The TF-IDF model, on the other hand, holds information on the more important words and the less important ones as well. It is rather easy to interpret bag-of-words vectors. But TF-IDF generally tends to perform better in machine learning models.

4 Dataset

We will be creating a custom made dataset for this chat bot, this is because we could not find a dataset online that suited our needs for this project. We will create and use the database in csv format for the TF-IDF chat bot and the same dataset in json format for the deep learning chat bot. Note that the two datasets are almost alike with the only difference being that the deep learning dataset has some extra question.

5 Real world applications

There are many applications of chat bot that will help customers customer journey smoother. Anyone in e-commerce will know the pain of losing prospects halfway through a marketing funnel. It doesn't take much to deter people from completing a purchase online, whether it's a confusing check-out system or hidden costs.

Buying a laptop is a big decision: You may end up using it for several years before getting another, and there are many makes, models, and chip configurations to choose from. also often a lots of people don't know exactly what they need, or what all the various hardware jargon means to solve this problem we have come up with this idea to create a chat bot that recommends it's users which laptop will be suitable for them. For example: This chat bot can be used by college students and other people to find affordable and good laptops. The chatbot suggests multiple laptops according to user's input. So the user will have multiple options to choose from. Using this chatbot people can easily find laptops according to their requirements. Also, there are many applications of chat bot which are given below:

5.1.Food Ordering
The second commonly visible application of chatbots is evident in the case of food delivery. Notable names such as Pizza Hut and KFC use chatbots for allowing customers to place orders through a conversation.

5.2.Companionship Applications
Chatbots could also be a game-changer in terms of companionship. The applications of chatbots as virtual and digital assistants could help in providing companionship to people in need, such as elderly people and Alzheimer's patients.

5.3.Healthcare Applications

The chatbot examples in the healthcare sector also showcase the breadth of the reach of chatbots. Chatbots, such as Super Izzy has been helping medical professionals in providing quick medical diagnosis and answers to health-related questions.

6 References

Twitter Bots and Gender Detection using Tf-idf Notebook for PAN at CLEF 2019. Mahmood, A., Srinivasan, P.(2019, September). Twitter Bots and Gender Detection using Tfidf. In CLEF (Working Notes).

Author Profiling: Bot and Gender Prediction using a Multi-Aspect Ensemble Approach Notebook for PAN at CLEF 2019. Giglou, H. B., Rahgouy, M., Rahgooy, T., Sheykhlan, M. K., Mohammadzadeh, E. (2019). Author Profiling: Bot and Gender Prediction using a Multi-Aspect Ensemble Approach.

Its All in a Name: Detecting and Labeling Bots by Their Name. Beskow, D. M., Carley, K. M. (2019). Its all in a name: detecting and labeling bots by their name. Computational and Mathematical Organization Theory, 25(1), 24-35.

Bots and Gender Profiling Using a Deep Learning Approach Notebook for PAN at CLEF 2019. Fontcuberta, J. R. P., De la Peña Sarracén, G. L. (2019). Bots and Gender Profiling using a Deep Learning Approach.

An Improved Text Sentiment Classification Model Using TFIDF and Next Word Negation.Das, B., Chakraborty, S. (2018). An improved text sentiment classification model using TF-IDF and next word negation. arXiv preprint arXiv:1806.06407.

Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents.Qaiser, S., Ali, R. (2018). Text mining: use of TF-IDF to examine the relevance of words to documents. International Journal of Computer Applications, 181(1), 25-29.

Hot Topic Detection Based on a Refined TF-IDF Algorithm. LI, Z., LIU, H. Y. G. Hot Topic Detection Based on a Refined TF-IDF Algorithm.

Unsupervised Sentence Representations as Word Information Series: Revisiting TF–IDF. Arroyo-Fernández, I., MéndezCruz, C. F., Sierra, G., TorresMoreno, J. M., Sidorov, G. (2019). Unsupervised sentence representations as word information series: Revisiting TF–IDF. Computer Speech Language, 56, 107-129.

Detecting Hate Speech and Offensive Language on Twitter using Machine Learning: An N-gram and TFIDF based Approach .Gaydhani, A., Doma, V., Kendre, S., Bhagwat, L. (2018). Detecting hate speech and offensive language on twitter using machine learning: An ngram and tfidf based approach. arXiv preprint arXiv:1809.08651

Research on Text Classification Based on Improved TF-IDF Algorithm. Fan, H., Qin, Y. (2018, May). Research on text classification based on improved tf-idf algorithm. In 2018 International Conference on Network, Communication, Computer Engineering (NCCE 2018). Atlantis Press.

Extractive based Text Summarization Using K-Means and TF-IDF. Khan, R., Qian, Y., Naeem, S. (2019). Extractive based Text Summarization Using K-Means and TF-IDF. International Journal of Information Engineering Electronic Business, 11(3).

Ranking of Text Documents using TF-IDF Weighting and Association Rules mining.Jabri, S., Dahbi, A., Gadi, T., Bassir, A. (2018, April). Ranking of text documents using TF-IDF weighting and association rules mining. In 2018 4th international conference on optimization and applications (ICOA) (pp. 1-6). IEEE.

Recurrent Neural Networks With TF-IDF Embedding Technique for Detection and Classification in Tweets of Dengue Disease. Amin, S., Uddin, M. I., Hassan, S., Khan, A., Nasser, N., Alharbi, A., Alyami, H. (2020). Recurrent neural networks with TF-IDF embedding technique for detection and classification in tweets of dengue disease. IEEE Access, 8, 131522-131533.

https://courses.analyticsvidhya.com/courses/Intro-to-NLP

https://dataaspirant.com/bag-of-words-bow/#t-1610979588750

https://landbot.io/blog/
natural-language-processing-chatbot
https://sloboda-studio.com/blog/
how-to-use-nlp-for-building-a-chatbot/
https://towardsdatascience.com/
how-to-build-a-chatbot-a-lesson-in-/
nlp-d0df588afa4b