Text Analytics an Approach to Artificial Intelligence

Vaishnavi D. Fate[1], Prof. A. B. Deshmukh[2], Prof. H. N. Watane[3]

Department of Information Technology, Sipna C.O.E.T, Amravati, Maharashtra, India[1]

Department of Information Technology, Sipna C.O.E.T, Amravati, Maharashtra, India[2]

Department of Information Technology, Sipna C.O.E.T, Amravati, Maharashtra, India[3]

Abstract

Text analytics supports organizations in managing unstructured information, identifying connections and relationships in information, and in extracting relevant entities to improve knowledge management activities. For the past decade, the amount of text messages sent monthly has increased by more than 7,700%. Younger generations overwhelmingly prefer texting to phone calls. And this is often scratching the surface, as there are many other sorts of textual data: support tickets, insurance application forms, healthcare records, product descriptions, and plenty of others.

Extracting meaning out of this text is an incredibly difficult task since texts may have different contexts and formats. Textual data is sometimes remarked as unstructured data because it doesn't have a transparent storage format or a predefined data model. Sure, you could put a sentence into an Excel cell. But how would that facilitate you to study it? The applications of text analysis are far and wide, from simple automation to advanced interactions between the person inputting the data and also the system they interact with. A fundamental example of that is a chatbot. This paper emphasis on how text analytics is a new approach to Artificial Intelligence.

Keywords: Text Analytics, Artificial Intelligence, Natural Language Processing.

Introduction

Text is one of the traditional ways of communication between people. With the growing availability of text data in electronic form, handling and analysis of the text by means of computers gained popularity. Handling text data with machine learning methods brought interesting challenges to the area that got further extended by the incorporation of some natural language specifics. As the methods were capable of addressing more complex problems related to text data, the expectations become bigger calling for more sophisticated methods, in particular a combination of methods from different research areas including information retrieval, machine learning, statistical data analysis, data mining, natural language processing, semantic technologies. Automatic text analysis has become an integral part of many systems, pushing boundaries of research capabilities towards what one can refer to as an artificial intelligence dream never-ending learning from text aiming at mimicking ways of human learning. Dunja Mladenic, Marko Grobelnik [1]

Text mining, sometimes alternately referred to as text data mining, roughly equivalent to text analytics, refers to the process of deriving high-quality information from text. High-quality information is typically derived through the divining of patterns and trends through means such as statistical pattern learning. Arturas Kaklauskas, Mark Seniut [2]

Text analytics is the process of transforming unstructured text documents into usable, structured data. Text analysis works by breaking apart sentences and phrases into their components and then evaluating each part's role and meaning using complex software rules and machine learning algorithms. Text analytics forms the foundation of numerous natural language processing (NLP) features, including named entity recognition, categorization, and sentiment analysis. In broad terms,

these NLP features aim to answer four questions: 1. Who is talking? 2. What are they talking about? 3. What are they saying about those subjects? 4. How do they feel?

Machine learning for natural language processing and text analytics involves using machine learning algorithms and "narrow" artificial intelligence (AI) to understand the meaning of text documents. These documents can be just about anything that contains the text: social media comments, online reviews survey responses, even financial, medical, legal, and regulatory documents. The role of machine learning and AI in natural language processing (NLP) and text analytics is to improve, accelerate and automate the underlying text analytics functions and NLP features that turn unstructured text into useable data and insights. Natural Language Processing (NLP) is the "ability of machines to understand and interpret human language the way it is written or spoken". The objective of NLP is to make computers/machines as intelligent as human beings in understanding language. The ultimate goal of NLP is to fill the gap between how humans communicate (natural language) and what the computer understands (machine language).

Literature review

When we speak about applying AI methods to text data, what we have got in mind is a whole range of methods and problems that in some way involve the analysis of text data. Many of those problems are addressed within the area of Text Mining. To be more concrete, we will briefly look at some example tasks that are addressed in our group during the last twenty years by applying AI methods to text data. These include 1. Visualization of text available in news articles, visualization of named entities over time, visualization of document corpus, visualization of websites. 2. Triplet extraction from text, document representation using semantic graph, document summarization. 3. Text enrichment, contextual question answering. 4. Semi-automatic ontology construction from document corpus, ontology extension. 5. Knowledge extraction from text, text mining combined with social network analysis. Arturas Kaklauskas, Mark Seniut [2]

Analysis of Academic Libraries' Facebook Posts: Text and Data Analytics analyzed a dataset of academic libraries posts on Facebook. It applied a text and data analytics approach to a dataset gathered from the Facebook posts of academic libraries at the highest 100 English-speaking universities, as listed by the 2014 Shanghai World University Rankings. The dataset is from a two-year posting history of 18,332 unique posts, 113,620 likes, and 3402 comments. But 1/4 of the libraries had more than 2000 post-related likes, and only seven received more than 100 comments on their postings. Content analysis identified the foremost frequent single word (unigrams), bigrams (two-word sequences), and trigrams (three-word sequences) in high and low engagement content. Sultan M. Al-Daihani, Alan Abrahams [3]

Broadcast interviews are the topic of study over some considerable time examining both news interviews further celebrity and talk show interviews. During this study we examine conventional genres and hybrid sort of broadcast interviewing using visual text analytic software Discursis. Discursis provides visual representations of whole interviews at a glance moreover because of the ability to focus on particular sections for closer analysis. Drawing on a corpus of 102 interviews from a single television program, this study check out if Discursis can relevantly visually represent forms of interviewing genres and highlights where shifting techniques are used within a single interview. Daniel Angus, Richard Fitzgerald [4]

Enterprise adoption of information technology (IT) innovations has been a subject of tremendous interest to both practitioners and researchers. This paper provides a complete integrative classification and analysis of the scholarly development of the enterprise-level IT innovation adoption literature by examining articles over the past three decades (1977–2008). We identify 473 articles and classify them by functional discipline, publication, research methodology, and IT type. The paper applies text analytic methods to the current document repository to (1) identify salient adoption determinants and their relationships, (2) discover research trends and patterns across disciplines, and (3) suggest potential areas for future research in IT innovation adoption at the enterprise level. Rahul C. Basole, C. David Seuss [5]

The GB railways collect about 150,000 text-based records every year on potentially dangerous events and therefore the numbers are on the rise within the Close Call System. The large volume of text requires considerable human effort to its interpretation. This work concentrates at visual text analysis techniques of Close Call records to extract safety lessons more quickly and efficiently. This paper treats basic steps for visual text analysis based on an evaluation test using a pre-defined test set of 150 Close Call records for ‘‘Trespass”, ‘‘Slip/Trip hazards on site” and ‘‘Level crossing”. The results demonstrate how new possibilities open up to develop interactive visualizations tools that allow data analysts to use text analysis techniques for risk analysis. Esteban, Peter Hughes [6]

Proposed work

Techniques like categorization, entity extraction, and sentiment analysis are used to identify insights, patterns, and trends in large volumes of unorganized data. Text analytics is an increasingly important task in marketing, as it can provide insights into a company’s text data. This study focuses on a developing Intelligence Platform that processes, analyzes, and provides insights around a company’s text data – i.e., surveys, call logs, social media posts, message boards, comments, etc. There are three key needs for using text analytics for market research 1 Built for surveys- A text analytics solution should be able to handle survey specifics, such people’s responses that fall into the ‘Nothing’ category, or ‘Other’. 2 Value-based pricing model. A text analytics solution needs a pricing model based on the value it delivers rather than the number of transactions. 3 Easy to use A text analytics solution must be able to be operated by an in-house data analyst (with little skills in NLP) who should be able to do occasional tweaks. Infotools[7]

The study on the related work done in this field shows that the models trained after extracting N-gram features from text give better results. The TFIDF approach on the bag-of-words features also shows promising results. Based on the review of features and the prominent classifiers used for text classification in the past work, we decided to extract N-gram from the text and weight them according to their TFIDF values. We feed these features to a machine learning algorithm to perform categorization. Given the set of tweets, this work aims to classify them into four section: hateful, offensive, clean, and average.

A. Data

The dataset that we have generated is a collection of three different datasets. The first dataset is publicly available on Crowdflower. This dataset contains tweets that have been manually classified into one of the following classes: “Hateful”, “Offensive”, “Clean”, and “Average”. The second dataset is also publicly available on Crowdflower, which consists of the tweets with the same classes as described earlier. The third dataset is published on Github and used in the work. Z. Waseem and D. Hovy [8] It consists of

two columns: tweet-ID and class. In this dataset, tweets corresponding to the tweet-ID are classified into one of the following three classes: “Sexism”, “Racism” and “Clean”.

B. Data Preprocessing

In the data preprocessing, we combine the three datasets. The task involves the removal of unnecessary columns from the datasets and enumerating the classes. For the third dataset, we recover the tweets corresponding to the tweet-ID present in the dataset. We use Twitter API for this purpose. The classes “Sexism” and “Racism” during this dataset are both considered as hate speech as per the definition. We convert the tweets to lowercase and delete the given subsequent contents from the tweets: Space Pattern, URLs, Twitter Mentions, Retweet Symbols, Stopwords. After combining the dataset in the proper format, we randomly shuffle and split the dataset into two parts: train dataset containing 70% of the samples and test dataset containing 30% of the samples.

C. Feature Extraction

We take out the n-gram features from the tweets and weight them consistent with their TFIDF values. The objective of using TFIDF is to bring down the effect of less informative tokens that appear again and again in the unstructured data. Observation is performed on values of n ranging from one to three. Thus, we consider unigram, bigram, and trigram features. The formula that is used to calculate the TFIDF of term t present in document d is:

$$\text{tfidf}(d, t) = \text{tf}(t) * \text{idf}(d, t)$$

Also, both L1 normalization and L2 (Euclidean) normalization of TFIDF is considered while performing experiments. L1 normalization is defined as:

$$vnorm = \frac{\text{v}}{|v1| + |v2| + \cdots + |vn|}$$

where n is the total number of documents. Similarly, L2 normalization is defined as:

$$vnorm = \frac{\text{v}}{\sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}}$$

D. Model

We consider three protruding machine learning algorithms used for text classification: Naive Bayes, Logistic Regression, and Support Vector Machines. We train each model on a training dataset by performing grid search for all the combinations of feature parameters and perform 10-fold cross-validation. The implementation of each algorithm is studied based on the average score of the cross-validation for each combination of feature parameters. The performance of these three algorithms is compared. Further, the hyperparameters of the two algorithms giving the best results are tuned for their respective feature parameters, which gives the best result. Again, 10-fold cross validation is performed to calculate the results for each combination of hyperparameters for that model. The model giving the highest cross validation accuracy is evaluated against the test data. We have used scikit-learn in Python for implementation.

E. Interfacing with twitter

Our final model is designed to interact with Twitter through the use of Twitter API particularly to collect data tweets via Twitter REST API. In python, we can use the Tweepy library which helps to add this functionality with simplicity. Twitter APIs, besides basic information such as the tweet text and the author of the tweet, return a data structure that contains additional information which can be used to provide

further analysis. For each maximum 140 characters tweet, API returns a JSON document containing several items of metadata presented as key and value pairs, out of which id and text are most important for the sake of this study.

We also create an application that acts as a module between the user and Twitter. Figure 1 shows the architecture of the application. Through our module, we can filter out hateful and offensive tweets being posted by an individual as well as classify the tweets posted on the user home timeline.

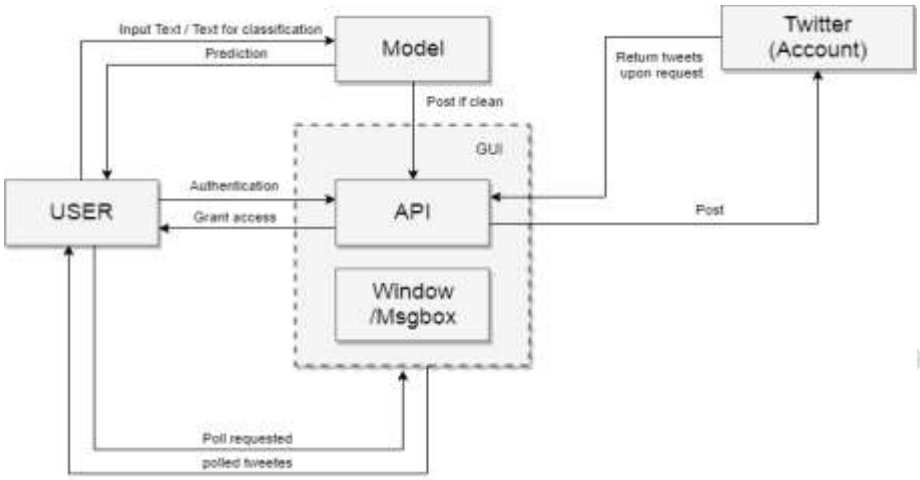

Figure 1: Architecture of the system interfacing with Twitter through Twitter API

Testing

Unit Testing

Unit testing is performed to test modules against detailed design. Inputs to the method are usually compiled modules from the coding process. Each module is assembled into a bigger unit during the unit testing process. Testing has been performed on each stage of project design and coding. We carry out the testing of the module interface to conform the right flow of information into and out of the program unit while testing. We ensure that the stored data maintains its integrity throughout the algorithm's execution by studying the local data structure. Finally, all error-handling paths also are tested.

System Testing

We usually perform system testing to search out errors resulting from unplanned interaction between the sub-system and system components. Software must be tested to find and rectify all possible errors once the source code is generated before delivering it to the customers. To find errors, series of test cases must be developed which will finally uncover all the possibly existing errors. We used different software techniques for this process. These techniques provide systematic guidance for designing a test that exercises the internal logic of the software components, exercises the input and output domains of a program to uncover errors in program function, behavior, and performance. We test the software using two methods: White Box testing**:** Internal program logic is exercised using this test case design technique. Black Box testing: Software requirements are analyzed using this test case design techniques. Both techniques help in finding the maximum number of errors with minimal effort and time.

Performance Testing

To test the run-time performance of the software within the context of an integrated system performance testing is done. These tests are implemented throughout the testing process. For example, the performance of individual modules is accessed during white box testing under unit testing.

Verification and Validation

The testing process is a part of a broader subject referring to verification and validation. We have to recognize the system specifications and try to meet the customer's requirements and we also have to verify and validate the product to make sure everything is in place. Verification and validation are two different things. One is performed to ensure that the software correctly implements a specific functionality and the other is done to ensure if the customer requirements are properly met or not by the end product.

Analysis and Result

Analysis

In this project, we propose an approach to devise a machine learning model which can differentiate between these two aspects of toxic language. We choose to detect hate speech, offensive language and clean text on Twitter platform. For that purpose, we use opensource Twitter datasets, we train our Logistic Regression, Naive Bayes and Support Vector Machines classifier model using n-gram and term frequency-inverse document frequency (TFIDF) as features and evaluate it for metric scores. We perform relative analysis of the results acquire using Logistic Regression, Naive Bayes and Support Vector Machines as classifier models. Our results show that Logistic Regression performs more effective among the three models for n-gram and TFIDF features after tune up the hyperparameters. We also make use of Twitter Application Programming Interface (API) to collect public user tweets from Twitter to spot tweets containing hate speech or offensive language. In addition, we create a module which serves as a mediator between the user and Twitter.

Results

The results of the relative analysis of Logistic Regression (LR), Naive Bayes (NB) and Support Vector Machines (SVM) for various combinations of feature parameters is shown in Figure 2 and TABLE 1.

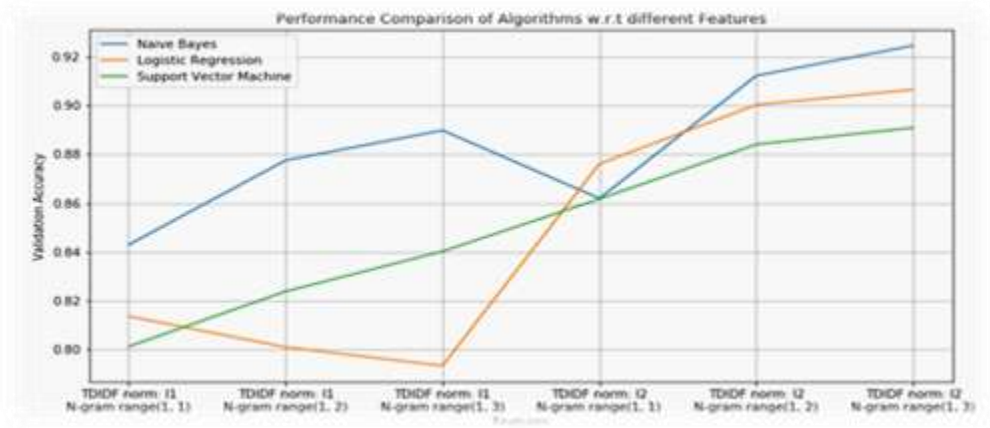

Figure 2: The figure shows comparative analysis of Naive Bayes, SVM and Logistic Regression on various sets of feature parameters

TABLE 1: Comparison of three models for different combination of feature parameters.

.

N-gram Range + TFIDF Norm	Accuracy		
	NB	LR	SVM
(1,1)+L1	0.842	0.816	0.802
(1,2)+L1	0.878	0.801	0.823
(1,3)+L1	0.890	0.794	0.841
(1,1)+L2	0.862	0.878	0.862
(1,2)+L2	0.913	0.901	0.884
(1,3)+L2	0.926	0.918	0.901

TABLE 2: Results after tuning Naive Bayes with respect to smoothing prior α for the features: n-gram range 1 - 3 and TFIDF normalization l2.

Alpha (α)	Accuracy
0.01	0.931
0. 1	0.934
1	0.925
10	0.877

TABLE 3: Results after tuning logistic regression w.r.t Regularization Parameter C and various optimization algorithms (solvers) For the features: n-gram range 1-3 and TFIDF normalization L2.

Regularization C + solver	Accuracy
10 + liblinear	0.949
10 + newton-cg	0.948
10 + saga	0.948
100 + liblinear	0.951
100 + newton-cg	0.950
100 + saga	0.950

TABLE 4: Classification scores obtained after evaluating the final Logistic Regression Model on test data.

	Precision	Recall	F-score
Hateful	0.94	0.96	0.95
Offensive	0.96	0.93	0.96
Clean	0.96	0.98	0.97
Average	0.96	0.96	0.96

TABLE 5: Confusion matrix for the evaluated test data on the final logistic regression model.

Class	Classified as		
	Hateful	Offensive	Clean
Hateful	0.965	0.021	0.014
Offensive	0.048	0.926	0.026
Clean	0.010	0.013	0.977

Conclusion

A solution to the recognition of hate speech and offensive language on Twitter through artificial intelligence using n-gram features weighted with TFIDF values. We performed relative analysis of Logistic Regression, Naive Bayes and Support Vector Machines on various sets of feature values and model hyperparameters. The results showed that Logistic Regression performs more effective with the optimal n-gram range 1 to 3 for the L2 normalization of TFIDF. Upon evaluating the model on test data, we achieved 95.3% accuracy. It was seen that 4.7% of the offensive tweets were misclassified as hateful. This problem may be solved by obtaining more samples of offensive language which does not contain hateful words. The results will be further improved by increasing the recall for the offensive class and precision for the hateful class. Also, it had been seen that the model does not account for negative words present in a sentence. Development can be done in this area by including linguistic features.

Reference

1] Dunja Mladenic and Marko Grobelnik," Automatic Text Analysis by Artificial Intelligence", Jozef Stefan Institute, Artificial Intelligence Laboratory, Jamova 39, 1000 Ljubljana, Slovenia.

2] Arturas Kaklauskas, Mark Seniut, Dilanthi Amaratunga, Irene Lill, Andrej Safonov, Nikolai Vatin, Justas Cerkauskas, Ieva Jackute, Agne Kuzminske, Lina Peciure, 'Text Analytics for Android Project", 4th International Conference on Building Resilience, Building Resilience 2014, 8-10 September 2014, Salford Quays, United Kingdom.

3] Sultan M. Al-Daihani, Alan Abrahams," Analysis of Academic Libraries' Facebook Posts: Text and Data Analytics", Department of Information Studies, Kuwait University, Kuwait Business Information Technology, Virginia Tech, United States.

4] Daniel Angus, Richard Fitzgerald, Christina Atay, Janet Wiles," Using visual text analytics to examine broadcast interviewing", School of Information Technology and Electrical Engineering, The University of Queensland, Brisbane, Australia

5] Rahul C. Basole, C. David Seuss, William B. Rouse, "IT innovation adoption by enterprises: Knowledge discovery through text analytics", Northern Light Group, One Constitution Center, Boston MA 02129, United States.

6] Miguel Figueres-Esteban, Peter Hughes, Coen van Gulijk," Visual analytics for text-based railway incident reports", University of Huddersfield, Institute of Railway Research, Queensgate, Huddersfield, UK.

7] case study, "Text Analytics for Market Research", Infotools, an international market research technology company.

8] Z. Waseem and D. Hovy, "Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter", Proceedings of the NAACL Student Research Workshop, 2016.