Bimodal Data Mining: Integration of Key Data and Semantic Analysis for Text/Audio Datasets

Victor Corja
Volgenau School of Engineering
Applied Information Technology, MS

Fairfax, Virginia
vcorja2@gmu.edu

Austin Crow
Volgenau School of Engineering
Applied Information Technology, MS

Fairfax, Virginia
acrow3@gmu.edu

Nannan Liu
Volgenau School of Engineering
Applied Information Technology, MS

Fairfax, Virginia
nliu6@gmu.edu

Abstract

There are 2.5 quintillion bytes of data created every day, with this amount of data it would be impossible to use it all without the assistance of tools and methodologies. Methodologies such as Speech Recognition, Textual Data Mining, and Semantic Analysis are used to help turn the 2.5 quintillion bytes of data into useful and usable data to help one's needs. Individually these tools and methodologies are incredibly helpful with anything from automated answering machines to determining the emotional implications of a given text. We aim to combine all of these individual methodologies into one consolidated model that will take data (speech or text), perform textual mining, and finally semantic analysis. First the models will be narrowed down by our criteria and corresponding sub-criteria to determine the strengths and weaknesses of each model. Once the models have been selected individual implementation is tested and the final model is created. This will then take the given data and give as much information as one could need in order to get a full understanding of the usefulness of this data. With proper implantation one will be able to gather information about where emergencies are occurring and who they pertain to, as well as determine political implications of news organizations across the nation.

Keywords— Data Mining, Speech Recognition, Text Mining, Semantic Analysis

I. Introduction

With the surge of social media, data creation is growing daily, and considering the amount of data available for use, there is a large gap between the potential usability and the current usage. It would be unfeasible for any team of analysts to manually look at and analyze all of the data, and effectively determine its possible uses, which has led to the growth in interest in textual and auditory data mining. Data mining is a term that has become more and more popular as technology becomes more integrated into our daily lives. Text-based data mining is a tool with an incredibly vast array of uses, from collecting specific filtered data representations, to identifying the preferences of particular groups of people based on metadata generated by their actions. A common use for textual data mining is a spam filter, which email providers implement to determine whether an email is likely to be spam, as opposed to having value to the user. These filters take into account not only the text of the message, but also user preferences and behaviors with respect to similar messages from the sender [1].

Aside from text-based data mining, the same technique can be applied to audio data. Ever since the creation of Siri for iOS in 2011, speech recognition technology has become more and more prevalent, especially as Amazon's Alexa and Google Home created a wave of mainstream attention into the technology. The uses for speech recognition are incredibly varied, and this includes implementations of data mining which can vastly improve the potential of data, as well as the generation of metadata, for audio resources [2].

In addition to data mining, semantic analysis can be applied to provide further insight into the data's meaning and potential. Semantic analysis enables a computer to derive the meaning of data and understand it with the context of its surrounding sentences, paragraphs, documents, and even entire datasets. Conversely, the same technique can be applied to the sentence and word

structures, as well as grammatical relationships, thereby providing an analysis both specific and broad in scope [3].

Individually all of these tools are great and incredibly useful. However, the authors believe that these tools can be used even better. By implementing one solid model that takes all of this raw data and goes through the speech recognition models, textual data mining, and finally semantic analysis to get the absolute most from the data. By taking this one implementation the understanding of the data would drastically increase and be able to help those in need if the data was emergency related or simply know how the polls are going in the case of political debate amongst the news. Throughout this paper we aim to help better understand what models were chosen as well as how this implementation will be used in the future.

II. Literature Review

There are a plethora of models that are used in order to pull information from any given data. But there are few that work together for a total data extraction to semantic analysis workflow [4]. The following sections outline models that are compared broken down into speech recognition, textual keyword mining, and semantic analysis.

In our literature review, we identified existing research into various methods of key data extraction and semantic analysis, for both audio and text resources. Our focus was on determining models we could integrate into our own research, and for this purpose we selected studies which utilized open-source models, and had significant documentation. In order to separate our analysis of the existing literature in our field of research, we split the review into three sections, each focusing on a particular aspect of our paper.

The first section is focused on publications which discuss models enabling the analysis of audio resources through the conversion of the audio data into text. Speech recognition is to convert a piece of the speech signal into text information. The system mainly includes four parts: feature extraction, acoustic model, language model, dictionary, and decoding. The final text after decoding will enable us to later perform keyword extraction and semantic analysis on the total combined data. Our initial research looked into the possibility of directly analyzing audio data, as well as converting the audio resources into text. While direct analysis would potentially be more effective for audio resources, considering the complexity of existing approaches, as well as the significant difference in both the requirements and the outputs of those approaches, we decided to focus only on integrating models dealing with speech recognition for audio-to-text conversion.

The second section identifies papers in which methodologies are proposed for the identification of keywords in text resources, such as topic mining and keyword extraction. While data analysis is usually performed on structured data, in order to retrieve all possible relevant information we will need to perform our analysis on the unstructured text data contained within our datasets. This will require data pre-processing in order to account for the enormous amount of information, followed by keyword analysis and topic mining.

The third section is focused on research into the semantic analysis of data, including the determination of key themes and moods within text and audio data. Data is composed of words, sentences, and paragraphs, so semantic analysis can also be divided into lexical-, sentence-, and paragraph-level semantic analysis to provide insight into the emotional and contextual composition of text resources. These analyses determine word sense disambiguation, the links between text entities such as location, time, and reason, and overall themes within the text, respectively.

Speech Recognition:

- There are many methods of performing speech recognition on audio data, a few of which are listed and ranked by Toshniwal [5]. The authors suggest that combining the traditionally separate automatic speech recognition (ASR), learned acoustic model, pronunciation model, and language model (LM) into the same single network is the best and most effective way of working with speech data. The focus of the paper is on the differences between the language models, which can be categorized into shallow, deep, and cold fusion, and have different integration timings and training times. Per the authors' analysis of the models based on tests performed on two datasets, it was determined that shallow fusion is generally the best approach until the "second pass rescoring", in which cold fusion takes the lead.

Hidden Markov Models (HMM)

- One of the oldest and most prevalent models for speech recognition, used for sequence analysis. Prior to the use of the model, feature extraction is required, primarily in the form of Mel Frequency Cepstral Coefficients (MFCC), since Markov chains require discrete states. After using MFCC, the extracted features are converted into discrete variables, and can be analyzed using HMM, which uses a generative probabilistic model to determine the next character based on the relationships between two sets of variables. In order to identify the next most likely character, the model requires an input with specific information about the language, which can be used for training purposes [6].

Listen, Attend, Spell (LAS)

- Listen, Attend, Spell (LAS) is an end-to-end model for automatic speech recognition, differing from HMM in that it does not make assumptions about the output sequence. LAS works by transcribing the audio sequence signal to a word sequence one character at a time. The operation is performed in two sequences, the "Listen" and the "Attend and Spell" operations. The first operation transforms the original signal into a high-level representation," while the second takes the high-level representation and produces the probability distribution over character sequences [7].

Recurrent Neural Network (RNN)

- The third model we looked at for speech recognition used Recurrent Neural Networks (RNN), a variation on standard neural networks focusing on differentiating phonemes. RNN does not require prior training in the language being analyzed, making it far more adaptable than models such as HMM. RNN focuses on networks with multiple feedback connections, creating nodes that contain deep-seated memory which can be queried [8]. By using backpropagation, each network can iterate through the provided data and create weights for likely values which can be shared across networks.

Textual Keyword Mining:

- The second section identifies papers in which methodologies are proposed for the identification of keywords in text resources, such as topic mining and keyword extraction. Keyword extraction algorithms are generally divided into two types: supervised and unsupervised. The supervised keyword extraction method is mainly carried out by classification, by constructing a vocabulary, and then judging the matching degree of each document with each word in the vocabulary, in a similar way of labeling. The advantage is that the accuracy is high, but the disadvantage is that a large batch of labeled data is required, and the labor cost is too high. Unsupervised methods have low data requirements. Currently, the commonly used keyword extraction algorithms are based on unsupervised algorithms. Such as TF-IDF algorithm, TextRank algorithm and topic model algorithm (including LSA, LSI, LDA, etc.). "Topic mining as a scientific literature can accurately capture the contextual structure of a topic, track research hotspots within a field…" [9]. By grouping key features from the data, the clusters then can be quantified in the number of relationships there are between topics and features, thus giving a strong visual analysis of what the data is about, all done with limited loss of the textual implications in the data.

Latent Dirichlet Allocation (LDA)

- Use the LDA model that comes with gensim. The principle of the usage method is: the candidate keywords and the extracted topics are calculated and sorted to obtain the final keywords. The key, how to calculate the similarity between candidate keywords and extracted topics? The idea is: each topic is represented by the set of N words multiple by probabilities. Each text belongs to k topics, and the words contained in the k topics are assigned to the document, and the candidate word keywords of each document are obtained. If the words obtained after document segmentation are among the candidate keywords, they are extracted as keywords. (Candidate keyword, generally refers to the word obtained after the document word segmentation, here refers to the word contained in the subject of the document).

Term Frequency-Inverse Document Frequency (TF-IDF)

- One commonly used approach to text analysis is topic mining. Another approach to text analysis, described by Lee and Kim [10], uses term frequency (TF) in metadata analysis, an identification of a word or word pattern that appears most frequently in the article, while ignoring common stop words - terms that do not add to the value of the article. The authors implemented an importance adjustment coefficient to measure whether a word is contextually relevant, using the Inverse Document Frequency (IDF) as the weight of the commonality of a term. The product of the TF and IDF is equivalent to the importance of the word within the article, and this method has the advantage of being simple and fast, and the result is more in line with the actual situation. The model involves a combination of topic mining and term frequency analysis and begins by creating a matrix of unique words within a dataset, then removing the highest and lowest frequency terms based on their TF-IDF product to account for both extremely common words like "and" and highly uncommon words. Once the matrix has been normalized to account for term frequency outliers, the authors then used the software R to randomly choose a distribution over topics and determine the topic proportions for certain topics within each document in the dataset. Each topic was randomly assigned to every nth word in each document, resulting in "the high-probability terms that define a topic in the corpus", and once the model had determined 20 topics for each document, the process was repeated

a total of 10 times, diminishing the impact of determining topics from a lower-frequency word.

TextRank

- The TextRank model is a graph-based ranking algorithm for determining term relationships within text-based datasets, based on Google's PageRank algorithm [11]. The modelling begins by displaying the significant words within the dataset as nodes, and creates edges between the nodes based on the degree of correlation within a close proximity. A TextRank score is then computed, based on the number of correlations and the significance decay by order of correlation, and the list of nodes with the highest scores is returned as the words of highest importance.

Semantic Analysis:

- Semantic analysis is the process in which a computer understands the sequence and meaning of words in the same way a human would, including a contextual understanding of colloquialism and homographs. In Wang, Wu, and Zhou's article they look into finding the reasoning for a high rate of registration but low rate of completion amongst Massive Open Online Courses. They use the Semantic Analysis Model (SMA) to track emotional tendencies of Learners in order to analyze the acceptance of the courses based on big data from homework completion, comments, forums, and other real-time information [12]. Semantic classification technology plays an important role in intelligent information processing services, identifying themes within the data and increasing the metadata which can be extracted from collected data. The sentiment being derived is emotional (Happy, Sad, Angry, Disappointed, Surprised, Proud, In Love, and Scared) this is all being done via a computer which inherently lacks emotion. Lexical semantics can use the characteristics of different content to classify lexical items. The task of semantic analysis is to conduct context-sensitive relation and classification reviews of data.

N-gram Model

- Tripathy et al. proposed the N-gram model, which presumes that the appearance of any given word is correlated with a selection of other words [13]. Using a set of words with a given length, the N-gram model attempts to determine the overall contextual sentiment based on the emotions contained within. A typical implementation process would break the given text into predefined sections by word length (grams), and analyze the individual contents, before proceeding to analysis with a gram of greater size. A typical example would be to analyze the sentence "The movie is not a good one."

    - Its unigram: "'The','movie','is','not','a','good','one'", would provide an overall positive result due to the presence of the word "good".

    - Its bigram: "'The movie','movie is','is not','not a','a good','good one'", which considers a pair of words at a time, would still provide the same result as the unigram.

    - Its trigram: "The movie is", "movie is not ", "is not a", "not a good", "a good one", which considers three words at a time, would provide an overall *bad* result, since it would take into account the presence of "not" before "good", and identify that as a negation.

Latent Semantic Analysis

- LSI looks for patterns in the way words cluster together to give further background meaning to particular clusters. This clustering is done through singular value decomposition (SVD) of the term-document matrix. The basic idea behind LSI is to take advantage of implicit higher-order structure in the association of terms with documents ("semantic structure") in order to improve the detection of relevant documents, on the basis of terms found in queries [14]. LSI keywords are related to the primary keyword, providing word sense disambiguation such that "iPhone" is a keyword of "Apple", while "Apple" is a keyword of both electronic products and fruits.

Convolution Neural Network (CNN)

- Emotion recognition is an important interdisciplinary research topic in the fields of neuroscience, psychology, cognitive science, computer science, and artificial intelligence. Convolutional Neural Networks (CNN) is a statistical learning model inspired by biological neural networks, the goal of which is to automatically mark the text with defined labels. Common text classification tasks include emotion recognition, email filtering, intent identification, and data classification. Two-dimensional signals such as image and voice are hard to be modelled well by traditional models like SVM, so the ability of CNN to characterize two-dimensional signals makes it far more usable in bimodal data analysis. CNN can also adaptively extract features to eliminate the dependence on human subjectivity or experience [15].

III. PROBLEM STATEMENT

The main challenge faced by data analysts after any significant event in which public sentiment is a key factor of analysis is the volume of information available. An occurrence affecting a large group of people, such as some form of a natural disaster, or an election, results in the creation of massive amounts of data without any standard format or medium, and this makes the job of the data analyst that much harder. We wanted to identify the most effective models currently developed for the semantic evaluation of text and speech data and provide a model of our own which would permit the input of any text and speech datasets, separately or combined, and return a comprehensive overview of the key data points and semantic identifiers contained within. This model would allow future researchers to bypass the issue of determining which medium to focus on, as the metadata analysis within would combine the efficacy of existing text and audio data parsing algorithms.

Our initial evaluation process was based on a literature review of existing proposals and ratings of models by prior researchers, which guided our decisions as to the capabilities built into our final implementation. Having determined the general composition of our model, we began the training process with a number of text and speech datasets of publicly generated data revolving around key events within the past decade, as well as a selection of product reviews which helped establish the baselines for semantic analysis due to the direct comparison with user ratings. Finally, we performed a comprehensive analysis of the capabilities and limitations of our model, as well as future work required to increase its effectiveness in a greater range of situations.

IV. PROPOSED METHODOLOGY

Prior to the final submission, the team's plan is to develop the final implementation of the integrated models selected and analyzed in milestone 2. Additionally, we will perform an analysis of the limitations of our final product, and recommendations for future work in the field.

Our proposed methodology consists of three key steps, starting with the identification of 9 existing models for key data and semantic analysis. These models will then be compared against one another in their respective categories. The models that perform the best will then move forward in our development process, and will be selected from:

- 3 speech recognition models
- 3 key data mining models for text data
- 3 semantic analysis models for text and audio data

The second step would be to perform tests on the chosen models, likely using scripts built in Python with a set of datasets including both text and audio data, which we would break up into 80% to be used as a training set, and 20% as an analysis set. Through this analysis, we would decide on the algorithms and methodologies we would use in our final proposed model. The third step would be to perform a complete analysis of a new set of datasets focused on a particular event, which would allow us to make an evaluation of our approach and determine its areas of strength and weakness. Once the best performing models are decided we will implement these models into a pipeline in which they all flow and work together. The model will be trained on the dataset via python, breaking up our set into a training set and test set, 20% and 80% respectively. Our dataset(s) will require a fair amount of pre-processing as they are likely to be formatted in dramatically different ways. The preprocessing will follow best practices depending on the models chosen. All pre-processing will be done keeping in mind not to change any of the actual data to keep sets valid and unbiased.

Through testing the accuracy of these models, we can determine what algorithms will likely be the best, and which models to base our deliverable on. After this testing is complete, we will take this model and perform an analysis of a new dataset that focuses on a particular well-known event, thus, allowing a proper evaluation of the approach. This will enable us to identify the comparative strengths and weaknesses of our approach, as well as the output differences versus the performance of the existing models.

V. RANKING MECHANISMS METHOD AND EVALUATION CRITERIA

Above we have determined three different models for speech recognition, text mining, and semantic analysis. In order to narrow down these models to the best from each respective category they were ranked based on a select criteria. These being compatibility, dimensionality, accuracy, and documentation (see Figure 1). These criteria were on a scale between one and ten, the total score was then used to determine what models would be used for the final implementation (see Figure 2).

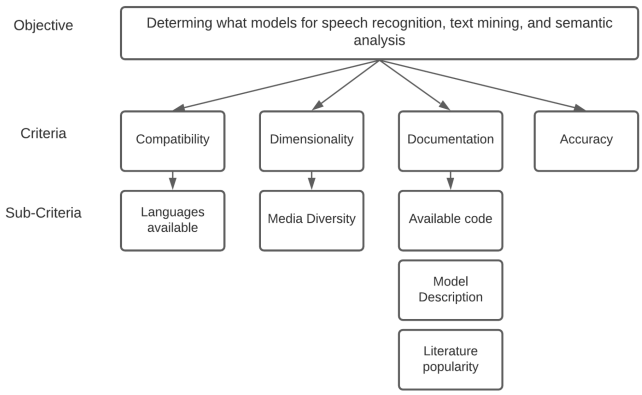

Figure 1: Analytical Hierarchy Process Diagram

- *Compatibility:*

This objective refers to how easily and with how much extra effort it takes to work with other models. This includes the language(s) in which the model can be used in as well as the amount of customization the model requires. The lowest ranking being "impossible to write this model in one of the languages we can use (i.e. Java, Python, R), or has some barrier to being used together with other models". With the highest ranking being "Easily works with other models and existing code for major languages (Python, R, JAVA)

- *Dimensionality:*

The dimensionality refers to being capable for both text and audio/ multiple functions. Such as keyword analysis and semantic analysis. The lowest score being "can only perform one function, and can be used on only one type of media". The highest score being "can do multiple kinds of analyses and can be used on multiple forms of media".

- *Accuracy:*

Based on a standardized dataset for both training and test data, how the models perform.

- *Documentation:*

This criteria refers to how much literature there is for the particular model as well as documentation on both how to build and how to use the models. The lowest score being "little to no documentation, no usable code, vague description for the model, not popularly used in literature". The highest score being "Plentiful and strong documentation, variety of existing code, detailed model description, and very common amongst academic literature".

		Compatability	Dimensionality	Accuracy	Documentation	Total
Speech Recognition	HMM	8	4	6	10	28
	LAS	7	4	8	7	26
	RNN	5	8	5	5	23
Textual Keyword Mining	LDA	7	8	7	6	28
	TF-IDF	7	5	4	7	23
	TextRank	4	7	6	6	23
Semantic Analysis	LSI	7	5	6	8	26
	N-Gram Model	4	3	6	7	20
	CNN	4	7	7	5	23

Figure 2: Analytic Hierarchy Process Scoring Table

VI. Models Chosen for Implementation

HMM

- Compatibility:
  - This criteria scored an 8 as it has been developed into working with many models and can be used in a variety of languages such as Python, R, and MatLab.
- Dimensionality:
  - A score of 4 was given as HMM primarily works with Speech recognition and text analysis and cannot be used for much more.
- Accuracy
  - A score of 7 was given for this model as when the data is pre processed accurately the results become very accurate.
- Documentation
  - The highest score was given for documentation as HMM was developed in the late 1900's and has an incredible amount of documentation and resources that refer to it.

LDA

- Compatibility:
  - A score of 7 was given for the implementation of the LDA process, because this model has been developed and used many times. Compared with the TextRank model, the LDA model has simple operation and fast calculation speed.
- Dimensionality:
  - A score of 8 was given to the LDA model. The traditional method of judging the similarity of two documents is to look at the number of words that appear in the two documents, such as TF-IDF. This method does not focus on semantic association. For example, "It's winter now", "Will summer clothes be discounted?" These two sentences do not have common words, but the two sentences are similar. If you judge the two sentences according to the traditional method, they are definitely not similar, so when judging the relevance of the document, you need to consider the semantics of the document, so LDA is one of the more effective models.
- Accuracy
  - The highest score was given for this model. The TF_IDF model means "The TF-IDF value increases when a specific keyword has high frequency in a document and the frequency of documents

that contain the keyword among the whole documents is low"[16]. For the same topic, the keywords may be the same, but due to the Inverse Document Frequency, the keyword score will not be very high, so the IF_IDF model is not a good choice. LDA can distinguish the same topic well and find out the keywords accurately.

- Documentation
    - A score of 6 was given for the documentation as the visualization of LDA has been greatly developed in the past ten years, so it has more research papers.

LSI

- Compatibility:
    - .A score of 7 was given as concise Python implementations are easy to find and many implementations have other functionality included (i.e. pre-processing and TF-IDF implementations)
- Dimensionality:
    - The middle score of 5 was given as LSI can only really be used on text, and generally only for getting main topics. It is not great for identifying emotional sentiments.
- Accuracy
    - A score of 6 was given for this model as LSI is accurate within reason to get the main topics of the data - not as accurate as latent Dirilecht analysis, and does not provide emotional data.
- Documentation
    - An 8 was received in documentation as there is a copious amount of data referring to theory however, not as much about how to actually implement the model.

VII. Implementation

For the implementation of our chosen models - HMM, LDA, and LSI - we chose to use existing code that we found on GitHub, as given our time constraints for this project, as well as the complexity of the models, we did not feel we would be able to write completely original code to implement all three models [17][18][19]. Our approach to the implementation was to identify code repositories that did not include an abundance of extraneous functionalities beyond what we desired for our final product, and that could work completely independently without dependencies on other repositories or non-standard modules. Given the prevalence of Python for the implementation of text- and audio-mining models, our final implementation was also written in Python.

The code implementation of the HMM model was based on the Python library scikit-learn, which provides a variety of functions for use with machine learning software [20]. The model identified phonemes (multi-letter units of speech) within the speech signals after accounting for noise within the recording, and applied an algorithm to determine the most likely word composed of the recorded phonemes.

The implementation of the LDA model used the Python gensim library for data preprocessing, as well as for the implementation of TF-IDF, on which the LDA model is partially dependent [21]. The gensim library is used widely for natural language processing (NLP) and facilitates the use of machine learning for unsupervised topic modeling. In our initial training phase, we used the code implementation in [18], slightly modified for our purposes, to analyze a dataset consisting of user reviews of products in Amazon's "Fine Dining" department. The model was able to generate the following topic compositions for the dataset, ranking from the highest probability to the lowest:

1. Score: 0.5362482070922852
   Topic: 0.014*"order" + 0.013*"amazon" + 0.012*"price" + 0.011*"store" + 0.011*"ship" + 0.010*"product" + 0.008*"great" + 0.007*"arriv" + 0.007*"purchas" + 0.007*"local"
2. Score: 0.2492866963148117
   Topic: 0.012*"water" + 0.007*"bottl" + 0.005*"tast" + 0.004*"drink" + 0.003*"like" + 0.003*"gummi" + 0.003*"product" + 0.003*"wine" + 0.003*"matcha" + 0.003*"good"
3. Score: 0.1334620863199234
   Topic: 0.010*"cereal" + 0.010*"butter" + 0.010*"popcorn" + 0.009*"peanut" + 0.009*"gluten" + 0.007*"free" + 0.007*"love" + 0.007*"oatmeal" + 0.007*"tast" + 0.007*"great"
4. Score: 0.06383361667394638
   Topic: 0.019*"food" + 0.019*"treat" + 0.015*"dog" + 0.010*"cat" + 0.010*"love" + 0.008*"chew" + 0.005*"train" + 0.005*"like" + 0.005*"chicken" + 0.005*"puppi"

The implementation of the LSI model was based on the same library as HMM [20], which made the final integration of the three models into a single working supermodel much simpler. By extracting the TF-IDF values for the text in the same Amazon dataset, we were able to train the model to recognize words that appeared at a higher frequency in positive and negative reviews, based on the accompanying rating given. Since all ratings were out of five, the model considered any reviews with a rating below 3 as negative, above 3 as possible, and excluded reviews with a rating of 3 to omit any neutral responses. The final output of the model provided an overall positivity and

negativity rating for the dataset as percentages, as well as generating a plot of the chi-squared distribution of words with the highest frequency in the dataset and their associated sentiment.

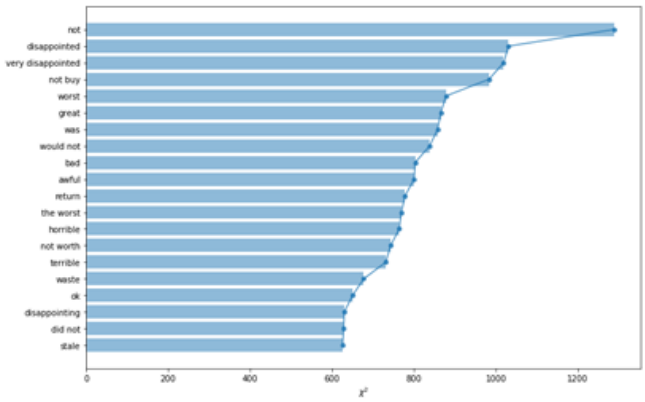

Figure 3: Chi-squared feature selection for Amazon review sentiments [19]

The final repository of code we obtained was able to determine whether the input data was a text or audio file, and either apply or omit the HMM speech recognition model to transcribe it depending on the determination. It then performed an analysis of the dataset's topic composition, as well as it's general semantic statistics, and presented this as a report, in addition to several other pieces of data derived during the script execution, such as the results of the TF-IDF calculations.

VIII. Conclusions

We will complete this section between milestone 1 and the final presentation, as it will require a more comprehensive analysis of the model and its limitations.

IX. Limitations

The discussion of the model limitations will be written together with the conclusions section, as it will require more investigation.

X. Future Work

This section will depend on the limitations we identify within our model.

References

[1] Wu, Yu, et al. "New anti-spam filter based on data mining and analysis of email security." Data Mining and Knowledge Discovery: Theory, Tools, and Technology V. Vol. 5098. International Society for Optics and Photonics, 2003.

[2] Tan, Zheng-Hua. "Audio and speech processing for data mining." Encyclopedia of Data Warehousing and Mining, Second Edition. IGI global, 2009. 98-103.

[3] Gautam, Geetika, and Divakar Yadav. "Sentiment analysis of twitter data using machine learning approaches and semantic analysis." 2014 Seventh International Conference on Contemporary Computing (IC3). IEEE, 2014.

[4] Shoumy, Nusrat J et al. "Multimodal Big Data Affective Analytics: A Comprehensive Survey Using Text, Audio, Visual and Physiological Signals." *Journal of network and computer* applications 149 (2020): 102447–. Web.

[5] Toshniwal, Shubham et al. "A Comparison of Techniques for Language Model Integration in Encoder-Decoder Speech Recognition." (2018): n. pag. Print.

[6] Chavan, Rupali S., and Ganesh S. Sable. "An overview of speech recognition using HMM." *International Journal of Computer Science and Mobile Computing* 2.6 (2013): 233-238.

[7] W. Chan, N. Jaitly, Q. Le and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4960-4964, doi: 10.1109/ICASSP.2016.7472621.

[8] Venkateswarlu, R. L. K., R. Vasantha Kumari, and G. Vani JayaSri. "Speech Recognition by Using Recurrent NeuralNetworks." *International Journal of Scientific & Engineering Research* 2.6 (2011): 1-7.

[9] Zhang, Tingting et al. "Multi-Dimension Topic Mining Based on Hierarchical Semantic Graph Model." *IEEE access 8* (2020): 64820–64835. Web.

[10] Sungjick Lee, and Han-Joon Kim. "News Keyword Extraction for Topic Tracking." *2008 Fourth International Conference on Networked Computing and Advanced Information Management*. Vol. 2. IEEE, 2008. 554–559. Web.

[11] Zhang, Mingxi et al. "An Empirical Study of TextRank for Keyword Extraction." *IEEE access 8* (2020): 178849–178858. Web.

[12] Wang, Ling, et al. "Semantic Analysis of Learners' Emotional Tendencies on Online MOOC Education." Sustainability, vol. 10, no. 6, 2018, p. 1921., https://doi.org/10.3390/su10061921.

[13] Tripathy, Abinash, Ankit Agrawal, and Santanu Kumar Rath. "Classification of Sentiment Reviews Using n-Gram Machine Learning Approach." *Expert systems with applications* 57 (2016): 117–126. Web.

[14] Zhang, Wen, Taketoshi Yoshida, and Xijin Tang. "A Comparative Study of TFIDF, LSI and Multi-Words for Text Classification." *Expert*

systems with applications 38.3 (2011): 2758–2765. Web.

[15] B. Zhang, C. Quan and F. Ren, "Study on CNN in the recognition of emotion in audio and images," 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), 2016, pp. 1 -5, doi: 10.1109/ICIS.2016.7550778

[16] Kim, SW., Gil, JM. Research paper classification systems based on TF-IDF and LDA schemes. Hum. Cent. Comput. Inf. Sci. 9, 30 (2019).

[17] wblgers, hmm_speech_recognition_demo, (2018), GitHub repository, https://github.com/wblgers/hmm_speech_recognition_demo

[18] bjherger, Easy-Latent-Dirichlet-Allocation, (2016), GitHub repository, https://github.com/bjherger/Easy-Latent-Dirichlet-Allocation

[19] susanli2016, NLP-with-Python, (2020), GitHub repository, https://github.com/susanli2016/NLP-with-Python

[20] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." *The Journal of Machine Learning Research* 12 (2011): 2825-2830.

[21] Rehurek, Radim, and Petr Sojka. "Software framework for topic modelling with large corpora." *In Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks*. 2010.

[22] Panayotov, Vassil, et al. "Librispeech: an asr corpus based on public domain audio books." 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2015.

[23] Ni, Jianmo, Jiacheng Li, and Julian McAuley. "Justifying recommendations using distantly-labeled reviews and fine-grained aspects." Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019.

Our Website: **AIT 582 Project Milestone 2 (gmu.edu)**