3-2021

Automated Sentiment Analysis for Personnel Survey Data in the US Air Force Context

Julia M. Haines

Automated Sentiment Analysis for Personnel Survey Data in the US Air Force Context

THESIS

Julia Haines, Civilian, USAF

AFIT-ENS-MS-21-M-164

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENS-MS-21-M-164

AUTOMATED SENTIMENT ANALYSIS FOR PERSONNEL SURVEY DATA IN THE US AIR FORCE CONTEXT

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

Julia Haines, BA

Civilian, USAF

March 2021

AFIT-ENS-MS-21-M-164

AUTOMATED SENTIMENT ANALYSIS FOR PERSONNEL SURVEY DATA IN THE US AIR FORCE CONTEXT

Julia Haines, BA

Civilian, USAF

Committee Membership:

Lance Champagne, PhD
Chair

Jason Freels, PhD
Member

Abstract

When surveys are distributed across the Air Force (AF), whether it be an employee engagement survey, a climate survey, or similar, significant resources are put towards the development, distribution and analysis of the survey. However, when open-ended questions are included on these surveys, respondent comments are generally under-utilized, more often treated as a source for pull-quotes rather than a data source in and of themselves. This is due to a lack of transparency and confidence in the accuracy of machine-aided methods such as sentiment analysis and topic modeling. This confidence reduces further when the text has special context, such as within the Air Force context. No model or methodology has been universally identified as ideal for this use case, nor has any model been universally adapted. The inconsistencies in approaches across analytical teams tasked with assessing the results of these surveys leaves data on the field.

This research quantifies the accuracy of some common sentiment analysis methods in order to gain a better understanding of the scope to which they should and can be applied. In order to investigate this question, various sentiment analysis packages and lexicons were implemented in R and applied to textual data from a survey distributed to Financial Management (FM) civilians across the Air Force Materiel Command (AFMC). Accuracy was assessed via comparison with manual sentiment classifications noted by a team of reviewers familiar with the FM career field and the Air Force context. The results indicate that sentiment analysis methods alone are not sufficient when applied to this context, although various adjustments were also investigated to significantly improve accuracy. This implies that AF analysts tasked with analyzing textual survey data should be hesitant to apply fully automated sentiment analysis as the sole method for generating conclusions about the body of text as a whole.

Acknowledgments

I would like to express my thanks and appreciation to my faculty advisor, Dr. Lance Champagne, for his guidance and support throughout the course of this thesis effort, as well as Dr. Jason Freels for his support. Their insight and experience was certainly appreciated. I would, also, like to thank my coworkers in A9A of Head Quarters Air Force Materiel Command, for their encouragement, support and guidance.

Julia M. Haines

Table of Contents

Page

List of Figures

Page

List of Tables

Automated Sentiment Analysis for Personnel Survey Data in the US Air Force Context

I. Introduction

General Issue

"Once upon a time, surveys were a staple for every leader to solicit feedback and every company to assess engagement. But now, surveys are starting to look like diesel trucks collecting dust in the age of electric cars." This quote comes from an article in Harvard Business Review titled "Employee Surveys are still one of the best ways to Measure Engagement," published in 2018 (Judd, 2018). The article is written by Scott Judd, head of People Analytics at Facebook, and Eric O'Rourke, People Growth & Survey Analytics lead at Facebook, a large company often noted for cultivating workforce climate in the environment of Silicon Valley. Despite the recent emphasis by business leaders on cultivating company climate and sustaining employee wellbeing, the article argues as its headline states, that employee surveys are an invaluable source of information.

Tracking employee attitudes across a workforce is not only worthwhile for cultivating climate, but also for predicting retention, maximizing profit margins and identifying markers of success for the company as a whole (Huselid, 1995). While some companies, in recent years, have opted to forgo the survey distribution, taking these metrics passively instead through tracking internet usage, email response and social networks, many studies have shown that the mere act of distributing a survey and asking employees for feedback has direct positive impacts on workforce health and unity (Judd, 2018). Additionally, when employees perceive that nothing is being done with or about

the results of a survey, there tends to be a more negative impact than if the survey had never been distributed at all (Council, 2020).

As a result, there is a lot of interest in how to efficiently and effectively analyze and act upon results of employee engagement surveys. Kenny argues in *Harvard Business Review* that "Managers, especially those in large organizations, spend an inordinate amount of time and money measuring the satisfaction levels of their staff. Sections of HR are dedicated to running employee satisfaction surveys and making sure managers conduct frequent check-ins with their direct reports." (Kenny, 2020)

In this vein, many business leaders default to asking and analyzing easily quantifiable questions with limited answers, such as multiple choice checkboxes and Likert Scales. These results are easy to interpret and lend themselves well to comparisons and charts. However, this leaves out perhaps the most insightful piece of an individual's survey response - their response to open-ended questions. This is the place they have to voice their opinions, feedback and suggestions for a company, unrestricted by "check those which apply" and hovering between "disagree" and "strongly disagree." (Jipa, 2019)

Even though these open-ended questions are often optional, response rates indicate that there is usually enough information to analyze without fear that a select few comments will carry the weight of an entire conclusion. For example, Facebook analysts recently wrote in Harvard Business Review that "when [Facebook] send[s] out a survey, we get a surprising volume of write-in comments: on average, 61% of our people submit their own feedback and suggestions, and each person touches on five distinct topics. It is clear that people take the survey seriously and want to be heard." (Judd, 2018)

Currently, typical analysis of these free-response questions is to focus on the frequency of buzzwords or to generate word clouds to depict what, overall, respondents said. This allows leaders to make statements like, “20% of people mentioned leadership when asked what they wanted to improve in their organization.” However, this is disjointed from the conclusion that is then drawn, which is that leadership is what needs to be improved upon. Suppose the question had asked “What do you want leadership to work on improving?” and “leadership” was mentioned so frequently in the context of “I want leadership to do xxx”? Strict reliance on frequency-based analysis methods paint conclusions with broad brushes, and ignore the intricacies of language which make it so difficult to study with definitive conclusions (Jipa, 2019).

Those who take the analysis a step further may attempt to expand the word cloud concept to topics instead, grouping synonymous words and like terms to represent key concepts, or topics, addressed in a piece of literature. This is called topic modeling and can be applied in both supervised and unsupervised facets. In some cases, it is easy to identify and define the topics one expects to see, and simply ask a model to diagnose the degree to which a paragraph aligns with those topics. In other cases, however, classifier and/or clustering methods may be used to identify those concepts and associated words, and the model may train itself on topics based on a subset of training data similar to or identical to the data itself (Qiu et al., 2009).

There is no clear “best” method for topic modeling or for sentiment analysis, since all models perform better and worse in different contexts, depending on the method itself and the data on which it was trained (Ribeiro et al., 2016). For example, a model trained on restaurant reviews may take a comment containing the word “fresh” and learn

to group it with like-comments, assumedly positive reviews about a salad or fish-based dish using similar words like “crisp”, “green”, and “organic”. However, if that same model is then expected to classify movie reviews, it would be incorrect to assume that comments containing the word “fresh” and “crisp” are of the same topic, since in that context, “fresh” likely refers to the plot or cast, and “crisp” the production film quality. Therefore, even though “fresh” and “crisp” have positive connotations in both contexts, the model will perform much better in the realm of restaurant reviews than movies reviews. This is important to keep in mind not only across topic categories, but also within topic realms wherein one category of commentary may span different languages, different dialects, and different time frames in which jargon has shifted (Jagtap & Pawar, 2013).

For this reason, analysts applying topic modeling and/or sentiment analysis methods to data must be aware of the original use-case scenario, and caveat those assumptions which may not translate well to new data. Sometimes, it is possible to adjust an existing model or method for a new use-case by re-interpreting (manually or automatically) a few select words that are highly context-specific (Tan & Zhang, 2008). One can also vary the application of multiple methods to achieve aspect-level sentiment analysis. However, in order to quantify if and by how much an adjustment improves the “fit” of the model, one must obtain an awareness of the performance of the model before and after the adjustment by quantifying the accuracy or notionally checking a few varied text pieces. In order to holistically measure the performance of a model, a substantial portion of the data needs to have been manually sorted or classified into the different sentiments and/or topics. Because this is not always feasible, however, analysts may

sometimes take a varied subset of the data, and if the results of the model seem to match or mirror notional logic, the application of the model, while not perfect, is deemed good enough.

Due to the widespread variety in the way that people talk, write, and even express sarcasm, and the room for different interpretations of a text even between two individuals, no model in linguistics will perform as well as models may in more predictable fields. Thus, "good enough" is relative. While far from perfect, the application of these models to long-form survey data may enable leaders to generate conclusions from employee feedback which are otherwise ignored, maximizing the return on investment in the survey and juicing the data for all it has to say. In the following experiment, advanced methods have been carefully applied and assessed for accuracy in an attempt to understand realistic accuracy expectations and identify shortfalls as potential areas of improvement. The result is a clearer understanding of which models and methods are capable of providing actionable, quantifiable and reasonably accurate conclusions drawn from previously muted qualitative linguistic data, specifically in the Air Force context.

Problem Statement

When surveys are distributed across and within the Air Force, whether it be an employee engagement survey or a climate survey or the like, many resources are put towards the development, distribution and analysis of the survey throughout its lifecycle. This is evidenced by the existence of the Air Force Survey Office. However, when open-ended questions are included on these surveys, respondent comments are under-utilized,

more often treated as a source for pull-quotes rather than a data source in of themselves (Jipa, 2019).

This is due to a lack of transparency and confidence in the accuracy of methods such as sentiment analysis and topic modeling when applied to the Air Force context. Often, these models are trained on and fit better with different categories of study, such as analyzing product reviews (Medhat et al., 2014). However, even when these methods are deployed, quantifying the accuracy demands hundreds of man hours spent manually reading through those comments and identifying topics mentioned and sentiments expressed. No model or methodology has been universally identified as ideal for this use case, nor has any model been universally adopted. The inconsistencies in approach across analytical teams tasked with assessing the results of these surveys leaves data on the field.

Research Objectives/Questions/Hypotheses

The purpose of this research is to develop a clearer understanding of how different sentiment analysis and topic modeling methods perform when applied to Air Force employee engagement survey data. Armed with this understanding, analysts of similar survey data across various career fields may be better equipped to transform previously underused qualitative respondent data into quantifiable, actionable insights with a degree of reliable accuracy and transparency.

Several different methods were applied to the data in an effort to assess their performance in this context. This research identifies the relative strengths and weaknesses

of each method to identify which approach performed the best, given the lexicon that it was trained on was most similar in structure and jargon to that found in the Air Force.

Organization

The remainder of the document is organized into three chapters. Chapter 2 reviews the relevant literature, chapter 3 details the steps and the methodology applied throughout this research, and chapter 4 summarizes the results and the conclusions which they indicate.

II. Literature Review

Data Pre-processing Steps

There are a number of pre-processing steps that are relatively constant across all linguistic analysis methodologies and applications. This includes steps such as stopword removal, stemming, Part of Speech (PoS) tagging, normalization, and tokenization (Clark, 2018). Below are definitions of basic principles in linguistic analysis.

Word normalization and lemmatization break down individual words to their lemma form, or common verb form, meaning that all instances of a root can be analyzed at the same level (Raja, 2017).

Stopword removal is a process that removes words that contribute little to the analysis. Often, this includes words such as “the”, “a” and “and”, since to analyze a textual piece and find that those are the most common, or are often associated with one another, would not be a novel finding. The list of stopwords is dynamic according to the application and the questions being asked by the analyst. In some cases, expressions of profanity may be removed, or replaced with appropriate counterparts. In other cases, such as in the research below, words such as “USAF” and “base” will be common throughout all comments, and may be removed in order to better emphasize more specific insights (Clark, 2018).

In some applications, the presence of non-textual but still linguistic indicators, such as emojis or emphasis on words, must be captured as more than a typo or an embedded image, since emojis and emphasis of the form “This brand is terrrrrrrrrrible” are often value-added and it would detract from the study to write them off as typos. In order

to break down a piece of text without taking away the meaning of these, there are a number of pre-processing steps that one must take. This includes tokenization, part of speech (PoS) tagging, and stemming, processes which are defined in further detail below (Carnahan, 2017).

Tokenization breaks a sentence into a list of tokens, which are most often words, but can also include punctuation, emojis and hashtags. PoS tagging uses context clues to identify the Part of Speech of those tokens, for example, a word ending in “ly” is likely an adjective. Stemming is a process which takes one instance of a word, and ensures that all variants of that word are included when considering frequency. This means that, when counting the instances of “happy” in a piece of text, the program is not only counting instances of “happy” alone, but also instances of “happier” and “happily”. Other approaches may use programs to generate two different types of textual matrices, one indicating TF-IDF features and the other indicating token occurrences (Borcan, 2020).

Stemming is the process by which instances of the same word in different terms will be considered effectively the same, allowing an algorithm to group “training”, “trains” and “trained” for context-specific questions. It is called stemming because the “ing” and “ed” are removed to reveal the source root word to be “train”. This can be done with verbs, nouns, and adjectives, with parsers available if a word can be used in multiple parts of speech (Clark, 2018).

In conjunction with these pre-processing steps, a piece of text may be re-formatted as a matrix of Term Frequency-Inverse Document Frequency (TF-IDF) values. This takes the frequency of a term in the immediate text being analyzed, and the frequency of that term in the general context, and computes a value reflecting how

important that word is to the piece of text (Clark, 2018). A word appearing frequently in the text, but which is also very common in all contexts, such as “the”, would have a lower value. However, a word very common in the text, but very uncommon in general, such as “dinosaur”, would have a higher value.

This matrix will also reflect which words appeared in conjunction with other words. For example, if the word “happy” was frequent in the text, but was also always preceded by “not”, it’s important to know not only the frequency of “happy” and “not” individually, but also how often they appeared next to one another. This is measured by an n-gram frequency. A bi-gram is the frequency of a two-word phrase, and a tri-gram reflects the frequency of a three-word phrase, etc. If negation is to be considered, such as allowing the first word in the phrases “not happy” or “no help” to cancel out the positive connotations of the second words, then a list of negatively connotated words must be identified (Pröllochs et al., 2018).

Sentiment Analysis

One way to adapt more complex analysis methods for survey analysis is to apply sentiment analysis, i.e. quantifying the degree to which a sentence or paragraph expresses positive, negative, neutral and/or other emotions/sentiments. This can be conceptualized as a classification and/or clustering method similar to the topic modeling described above, but instead of grouping aspects by topic association, they are instead grouped by sentiment association, whether it be positive and negative or, as found in the National Research Council Canada (NRC) Emotion Lexicon, association with 10 core emotions such as trust and anger (A. & Sonawane, 2016).

This can be done at multiple levels, depending on if the analyst is interested in whether the paragraph as a whole trends positive, whether a sentence trends positive, and even whether mentions of a topic within the sentence trends positive. The scale of the score differs by method, but for example, scores of +1 indicate positive sentiment, scores of -1 indicate negative sentiment, and scores near or at 0 indicate neutral sentiment. When these scores are computed at the topic level, it is called *aspect-level sentiment analysis*, a combination of topic modeling and sentiment analysis applications (Luo et al., 2016).

Most algorithms related to sentiment analysis can be categorized as either rules-based, automatic or a hybrid. Rules-based algorithms are somewhat analyst-friendly, meaning that the analyst can determine the values of parameters and adjust the algorithm for their use-case. At the most basic level, rules-based algorithms are built off of initial datasets, which use two lists of "positive" and "negative" marked words to identify those sentiments in a text.

Beyond the initial assessment, which may consider the number of positive or negative words to define overall sentiment, this approach can be catered using methods such as Part of Speech (PoS) tagging, tokenization and stemming (Staff, MonkeyLearn, 2020). "We can combine any of the machine learning techniques with natural language processing (NLP) techniques like Hidden Markov Models (HMM), N-gram, POS, Bag of word and large sentiment lexicon acquisition for better and accurate results for implicit and explicit sentiment analysis." (Staff, MonkeyLearn, 2020)

Support Vector Machines (SVM) and N-gram algorithms are used together for emotion identification of twitter messages (Almatarneh & Gamallo, 2019). Rules-based

algorithms are generally easy to follow and easy to implement, but they are difficult to maintain since the rules need to be updated with a degree of consistency and the analyst needs to have a very active role in defining the rules being used. The other type of sentiment analysis algorithm, automatic algorithms, take longer to set up and train since they are based in machine learning, but are often more accurate and holistic in their results (Almatarneh & Gamallo, 2019). In order to first use the data to train the algorithm, it is fed through an n-grams or bag of words type process so that the machine can identify factors of the string which may contribute to the string's sentiment score (Almatarneh & Gamallo, 2019).

Approaches can also be categorized into both supervised or unsupervised methods. Supervised methods require the analyst team to either find a dataset with associated polarity scores (such as Amazon reviews tied to a number of stars for a product) or they must manually label the polarities in order to give the model a training set on which to base predicted scores. Supervised methods include Naive Bayes (NB), SVM, K-Nearest Neighbor (KNN) and Maximum Entropy (ME). There are a large number of options for training sets to pair with supervised methods, such as the Stanford Sentiment Treebank (SST-5), which was created through Amazon Mechanical Turk (Meza, 2015). The most common single-word classifiers include the NRC, Bing and AFINN lexicons, as well as SentiWordNet (Staff, MonkeyLearn, 2020). Manually-assigned scores for each word in these lexicons are built off of human input, but it has been shown that statistics-based lexicons perform better than individual human scores, as shown in T. (Pang & Lee, 2008)

Unsupervised methods may depend on neural networks to identify the sentiment of different points, or instead use k-means clustering, hierarchical clustering and semantic orientation. Both can be paired with Term Frequency and Inverse Document Frequency (TF-IDF), PoS tagging, negations, dependencies and opinion words and phrases through machine learning and supervised methods (A. & Sonawane, 2016). If this process is done manually, i.e. unsupervised, it is time consuming and laborious, as the analyst must manually extract and identify "opinion words" from the document (Varghese & M, 2013). Other unsupervised methods include Hidden Markov Models (HMM), Neural Networks, Principal Component Analysis (PCA), ICA and SVD (Jotheeswaran, 2012; Varghese & M, 2013).

Mathematical Basis of Common Approaches

There are many schools of thought about the correct approach to accurately assess the sentiment of a given piece of textual data. Table 2 displays the strengths and weaknesses of four of the more common approaches; linear regression, logistic regression, Naive Bayes and KNN. Sometimes, a combination of approaches is the recommended best approach (Samuel et al., 2020).

Table 1: Generalized Sentiment Analysis Approaches (Samuel et al., 2020)

Classifier	Characteristic	Strength	Weakness
Linear regression	Minimize sum of squared differences between predicted and true values	Intuitive, useful and stable, easy to understand	Sensitive to outliers; Ineffective with non-linearity
Logistic regression	Probability of an outcome is based on a logistic function	Transparent and easy to understand; Regularized to avoid over-fitting	Expensive training phase; Assumption of linearity
Naïve Bayes classifier	Based on assumption of independence between predictor variables	Effective with real-world data; Efficient and can deal with dimensionality	Over-simplified assumptions; Limited by data scarcity
K-Nearest Neighbor	Computes classification based on weights of the nearest neighbors, instance based	KNN is easy to implement, efficient with small data, applicable for multi-class problems	Inefficient with big data; Sensitive to data quality; Noisy features degrade the performance

The Naive Bayes classifier (NBC) uses Bayes Theorem to determine the class of a piece of text based on the highest conditional probability, which is calculated with maximum *a posteriori* estimation (Samuel et al., 2020). It can be effectively applied to shorter pieces of text such as tweets, but it is a probabilistic classifier, and is most often used at the document level. “Naive Bayes is optimal for certain problem classes with highly dependent features. Naive Bayes classifiers are computationally fast when making decisions. It does not require large amounts of data before learning can begin.” (Moralwar & Deshmukh, 2015)

The multinomial Naive Bayes approach and Bernoulli Naive Bayes approach represent the features of a document in a binary fashion and a frequency-based fashion, respectively. “A comparative study showed that NBC has higher accuracy to classify documents than other common classifiers, such as decision trees, neural networks, and support vector machines.” (Samuel et al., 2020). In general, machine learning techniques

such as Naive Bayes, SVM and ME are more accurate in sentiment classification. One of the benefits of NBC is that it does not require a huge training set of data, and it is relatively efficient, but the assumptions that it makes about conditional probability and distribution types may be over-simplified (Samuel et al., 2020). Additionally, in practice, Bayes Classifiers are computationally expensive to train, so while the decision making may be fast, the upfront costs mean it is not always the best choice.

The Logistic Regression classification method can be applied to longer tweets, and is considered a discriminative classifier. As one of the older methods for sentiment classification, it uses a logistic function to minimize error, and it has been shown to have a higher degree of accuracy than NBC, SVM, Random Forest and Decision Tree methods (Samuel et al., 2020).

Accuracy can be increased if stepwise logistic regression methods are employed. However, the "stability of the logistic regression classifier is lower than the other classifiers due to the widespread distribution of the values of average classification accuracy" and "LR classifiers have a fairly expensive training phase which includes parameter modeling with optimization techniques (Samuel et al., 2020)." When comparing NBC to Logistic Regression, it has been found that NBC performs better for both smaller tweet lengths and longer tweet lengths. (Kiprono & Abade, 2016).

SVMs are a type of linear classifier categorized under supervised machine learning. With countvectorizer numeric matrices and TF-IDFs, with a weighting scheme developed specifically for unigrams, one can train the algorithm to classify tweets in a more accurate manner than NBC (Almatarneh & Gamallo, 2019). SVM are trained through the development of a pattern recognition technique, minimizing the probability

of error and building a hyperplane, bringing the SVM to become a quadratic optimization problem (Almatarneh & Gamallo, 2019).

In SVM, a hyperplane is built with text examples as data points in a multidimensional space. Specific areas of the space represent different sentiments, and text points introduced to the set are given a category based on clustering with existing points and the relative regions in the plane (Staff, MonkeyLearn, 2020). This technique for news articles and blogs, if the user wants to identify positive, negative and neutral examples, tends to perform well (Almatarneh & Gamallo, 2019).

K-Nearest Neighbor is a classifier that categorizes a text object based on that object's k-nearest neighbors. Training documents, which are similar in structure to the test document, are given category labels to build the set. Both Euclidean and Manhattan distances can be used for computation, as could many other mathematical norms, and no computation is done until there is a need for classification. This is known as a "lazy learning" function. Chebyshev Norm and Mahalanobis distances may also be used, but Euclidean and Manhattan are most common (Dua, 2020). Most measures of distance would be appropriate, given they are computed consistently. Classification is based on the nearest neighbors of the new data point, and the user may set "k" to identify how many nearest neighbors to include, or the user may identify a radius and all neighbors in that radius will be included (Dua, 2020).

Maximum Entropy is a probabilistic classifier that does not rely on assumptions about independence in a data set. The goal of this technique is to "maximize the entropy of the induced distribution subject to the constraint that the expected values of the feature/class functions with respect to the model are equal to their expected values with

respect to the training data: the underlying philosophy is that we should choose the model making the fewest assumptions about the data while still remaining consistent with it, which makes intuitive sense (Jagtap & Pawar, 2013)."

Other approaches have been used, including the Winnow technique (mistake-driven weight vectoring), Association Rule Learning, Semantic Orientation Approach, Mutual Information (MI), Residual Inverse Document Frequency (RIDF), TF-IDF and Decision Tree Learning (DT) (Moralwar & Deshmukh, 2015; Tan & Zhang, 2008). The most common algorithms for Decision Tree Learning are ID3, C4.5 and CART (Moralwar & Deshmukh, 2015). Additionally, some approaches take into consideration the personality of the person writing the text, specifically by using DISC (Dominance, Influence, Compliance and Steadiness) assessment techniques. "DISC assessment is useful for information retrieval, content selection, product positioning and psychological assessment of users. A combination of psychological and linguistic analysis was used in past research to extract emotions from multilingual text posted on social media." (Samuel et al., 2020)

This research will focus on sentiment classification techniques which are based on a variety of lexicons, since these methods are the easiest to apply and the easiest to explain when creating presentations for leadership. They are also the most widely-available, and don't require too much manual work on the part of the analysts, i.e. there is a lower set-up investment in these techniques. Techniques which require building a new lexicon by-hand, or training a new model on unique, unreplicable data, would be difficult to implement across a variety of applications and would not translate well between fields even within the Air Force context, while introducing fears of overfitting and wide

assumptions. Therefore, in this research, more common, generalized approaches were tested.

Limitations

Identifying the scope that the researcher is interested in is often the first hurdle, since to answer a simple question such as "how do people feel about the movie" means not only looking at user reviews on Rotten Tomatoes, but also considering a variety of other sites, Amazon reviews of related products, chats in online forums, blog posts relating to the movie, and the search could continue. Often, then, it seems that studies tend to limit their scope to a single website whose purpose is to provide consumers a platform to state their opinion, which in turn narrows the initial question asked.

Additionally, as is a problem with all degrees of text analysis, a writer's choice to quote the opinion of another presents an obstacle, since one must determine whether the citation is from a place of approval or disapproval, or to disregard the reference all together and focus on the original content of the review. Then, even once the sentiment is identified, one may be inclined to pair opinion mining with topic classification methods to find out which character or which aspect of the movie, for example, is turning people away. As is pointed out in Source A, options for summarizing the sentiment of a data set include the following main choices: "(a) aggregation of "votes" that may be registered on different scales (e.g., one reviewer uses a star system, but another uses letter grades) (b) selective highlighting of some opinions (c) representation of points of disagreement and points of consensus, (d) identification of communities of opinion holders , and e) accounting for different levels of authority among opinion holders" (Pang & Lee, 2008).

Additionally, the context or topic of reviews may give different meanings to the same expression. For example, the sentence “it doesn’t taste like anything” may be associated with a positive review for a protein powder, but a negative review for a salt free potato chip. Additionally, the ability to distinguish between a fact and an opinion represents another hurdle, since some unfortunate facts would likely register negative even if the reviewer has a positive impression despite a negative fact. Here, the order of the statements also matters when assessing the overall sentiment of a review. For example, if a review of the latest Toy Story stated “Pixar has changed since being acquired by Disney, it was a good movie.” that expresses a different overall sentiment than “It was a good movie, but Pixar has changed since being acquired by Disney.” (Jipa, 2019)

There are a large number of approaches one can take to begin assessing the sentiment contained in a text document or dataset containing data of a linguistic nature. Therefore, before choosing an approach, the researcher must identify the questions they want to ask, the degree of accuracy versus ease that is realistic for them to achieve, the availability of training data sets (do they have to build one specifically for their cause), and whether they are only interested in positive versus negative sentiment, or whether they want a more detailed analysis (Kiprono & Abade, 2016)

Sentiment Classification Methodologies

Because there are a wide variety of applications of sentiment analysis, there are also many different methods that are available to use, depending on the dataset and use case. There have also been studies assessing the differences between these methods,

which have proven that there is no one “best” application. “The benchmark analyses reveal that there is no superior sentiment analysis method because all tools perform differently depending on the specific context they are applied on or depending on the corresponding data source on which they were trained.” (Feine et al., 2019) Thus, an ideal sentiment analysis method must be selected not only based on the data with which it was trained, or based on, but also based on the perceived or calculated accuracy of the method when applied to specific data.

A study titled “SentiBench - a benchmark comparison of state-of-the-practice sentiment analysis methods” published by EPJ Data Science Journal in 2016 found that two of the “best” methods for measuring numerical polarity in sentiment analysis, identifying positive, neutral, and negative comments, were VADER and AFINN. Both use a set of rules and heuristics to assess the degree to which a piece of text aligns with a given lexicon, and those lexicons are trained on social media data (Ribeiro et al., 2016).

Different, machine-learning based approaches developed by technology companies such as IBM, Microsoft and Google have been shown to perform better on varied datasets. With these methods, machine learning classification algorithms are used to predict the sentiment score of a piece of text. Thus, for this research question, the survey results can be applied to both rule-based and machine learning-based methods through open-source APIs (Corredera et al., 2017) and the webservice ifeel 2.0 (Araujo et al., 2016), as they did in the study of Chatbot Customer Service Sentiment Analysis (Feine et al., 2019).

In order to properly compare these methods, this research will standardize the sentiment scores obtained by each of these methods, and conduct correlation tests

between the sentiment scores of different methods with those computed manually by the team of analysts in HQ AFMC. This will reveal which, if any, of the sentiment analysis methods are valid for application to this type of data.

Improvement on SA with Pronouns

Many sentiment analysis approaches immediately conduct a pre-processing step called “stopword removal”. In this step, all words considered neutral are removed from the lexicon, to include words such as “and’, “the”, “as” and “to”. There is no universal list of agreed-upon stopwords, since the stopwords can be context-dependent. Common lists pull from onix, snowball or SMART. For purposes of applications, here, this research used a conjugated list from the three sources. The Onix list, the list derived from the System for the Mechanical Analysis and Retrieval of Text (SMART) Information Retrieval System developed by Cornell University in the 1960s, and the Snowball Stopword List (Salton & Buckley, 2019.; *SMART Stopword List*, 1960; *Snowball Stopword List*, 1979.; *Stopword List 1*, 2001)

Improvement on SA with Context Words

In 1997, a study revealed a method to define polarity of new words using connector pairing words such as “and”, “but”, “either or” etc. For example, say a model has no knowledge of the polarity of the word “productive” or “boring” but it knows that “great” is positively associated. If a sentence said “The office I work in is great and productive.” then the connector “and” would teach the model that “productive” likely trends in the same direction as “great”, i.e. positively. However, if the sentence said “The office I work in is great but boring.” then the model would use the “but” connector and

learn that the new word trends in the opposite direction of "great," i.e. negatively (Hatzivassiloglou & McKeown, 1997). Further work learned to pull data from online dictionaries and thesauruses to define sentiment words, and used a variety of statistical approaches to measure the "distance" from a new word to original, pre-defined sets of seed words to represent good and bad. However, this method does not do well when applied to domain-specific, infrequent sentiment words. (Qiu et al., 2009)

Improvement on SA with Negation Phrases

In 2019, Jipa investigated several different approaches to perform the classification using different text features: unigrams (individual words), selected words (such as verb, adjective, and adverb), and words labeled with part-of-speech tags. Then, product reviews were analyzed to identify the sources of error and directions for improving the performance of the SVM classifier. The second part of Jipa's study investigates the use of negation phrases through simple linguistic processing to improve classification accuracy (Jipa, 2019; Na et al., 2004).

Improvement on SA with Double Propagation Method

"In most sentiment analysis applications, the sentiment lexicon plays a key role. However, it is hard, if not impossible, to collect and maintain a universal sentiment lexicon for all application domains because different words may be used in different domains (Qiu et al., 2009)." Here, simply stated, it is noted that domain and context play key roles in distinguishing the application of sentiment analysis techniques. The lexicon, or dataset, that a model is trained on, or on which a methodology is tested, largely influences the realm of sets to which that model may be re-applied with any degree of

confidence in the approach. In a 2006 study at Zhejiang University and University of Illinois at Chicago, two students attempted to solve this problem with a technique called Double Propagation. Rather than taking sentiment of words from a multi-domain corpus, it takes into account the context in which words appear in the corpus to which the method is being applied (Qiu et al., 2009).

In their research, they attempt to identify words specific to a particular domain using a small set of seed sentiment words. Then, domain specific words are extracted using identifiable features in the text, and keeps feeding itself until no additional features or sentiment words are identified. Dependency grammar is used to identify these features and respective sentiment words, and then predict polarities of newly-defined words. Previous methods for this are explained below. For this task of double propagation, there are four main steps (Qiu et al., 2009)

1. Extract sentiment words using sentiment words
2. Extract features using sentiment words
3. Extract sentiment words using features
4. Extract features using features

Minipar is used to parse the sentences, and the Stanford PoS tagger is also deployed.

Topic Modeling with Latent Dirichlet Allocation (LDA)

In this technique, comments are accumulated and treated as a bag of words, each with different probabilities, and then topics are derived that compose each comment. The assumption is that topics and words each have distributions underlying the text, and one can use those distributions to identify topics and the words associated with them. LDA is

the most common approach, but requires the text to be transformed into a document term matrix and cleaned for punctuation, etc, before being applied (Koch, 2020).

In this research, LDA was applied with the package Gensim in Python, as well as with R. Essentially, the analyst must decide to what degree she wants the topic modeling to apply, how many words should be associated with each topic. This restricts the algorithm from identifying all text in the corpus as under one topic, or from splitting it into as many topics as there are comments. The ideal number for this choice depends on the size of the text and the questions being asked by the analyst. Next, the analyst may choose a topic mixture, i.e. what they expect the degree of topics to be distributed among those identified. Then, words start mapping to the topics and the model starts to learn (Chen, 2011; Clark, 2018; Wang, 2017).

Aspect-Level Sentiment Analysis with OLS Regression

Luo, Zhou and Shon conducted text analysis on employee reviews about top Fortune 500 companies posted on career information site Glassdoor. They used a previously defined framework that categorizes the text into 9 corporate-advertised values: Integrity, Teamwork, Innovation, Respect, Quality, Safety, Community, Communication, and Hard Work, with keywords associated with each provided (Guiso et al., 2012). They then conducted standard text processing and data cleaning to reduce the effect of noise, which included the removal of stop words and stemming of key words. Next, using bag-of-words, they extracted term frequencies for each of the categories and used that to perform sentiment analysis to assign polarity for each review (Guiso et al., 2012).

This approach can be modified if one adjusts the 9 categories to fit those of the themes, or of the subtopics, identified in the dataset, with associated keywords, and one can conduct analysis about the performance of this approach using the manual categorization available to us. Similar to how they used a financial measure of success for respective companies, one can instead use the overall sentiment score identified as the dependent variable in ordinary least squares (OLS) regression tests. One could also introduce factors like length of response to the regression to see how that may impact the overall sentiment of the comment. (Luo et al., 2016)

Aspect-Level Sentiment Analysis with SVM

Costa and Veloso used machine learning classification algorithms to identify sentiment in employee reviews, such as Support Vector Regression (SVR) and SVM. It converted all reviews to a term frequency inter document frequency (TF-IDF) matrix with respective weights, and used classification algorithms like SVMs to separate the sentiments identified within the text.

However, shortfalls of this method ignore frequencies of related synonymous words. Therefore, this research not only used classification SVM methods to assess sentiment of employee reviews, but also evaluated the application of approaches that take into account the vector representation of related words as they appear in similar contexts in common spaces. Then, this modified the SVM approach to better represent the text. They pulled data from Indeed and LM. “Labeled reviews from Indeed come with ratings (ranging from 0 to 5) based on management, culture, work/life balance, benefits, and

career opportunities." They then used RMSE to assess accuracy, with 10-fold Cross Validation (CV).

"We collected a large number of job reviews posted in social platforms, as well as survey data such as work/life balance, management, culture, and also official data about retention and salary. We performed a systematic set of experiments in order to evaluate our proposed sentiment analysis approaches. … Specifically, we used the SVR algorithm for regression, and the SVM algorithm for classification. These algorithms follow a supervised learning strategy, and associate patterns in the vector representation of the review and a variable or criterion of interest. Criteria can assume values as salary, retention, management, culture, work/life balance, and others. … To evaluate the prediction performance of our approaches, we have used the standard Root Mean Squared Error (RMSE) measure, which gives a summarized measure of the prediction error for regression tasks, and the standard accuracy and F1 measures for classification tasks. We conducted tenfold cross validation using Indeed and LM datasets." (Costa & Veloso, 2015; Lu et al., 2016.; Salas-Zárate et al., 2017)

III. Methodology

Assumptions/Limitations

As with all linguistic analysis methodologies and applications, there are substantial assumptions which should be stated prior to definitive conclusions being drawn. Below, those assumptions have been generalized, but more specific assumptions relating to certain methodologies or mathematical principles are detailed at greater length in "Methodology".

Sentiment analysis techniques are notorious for their inability to accurately and consistently detect sarcasm (Salas-Zárate et al., 2017). For example, take a respondent who writes "I think leadership is doing a great job, I really love how considerate they are of the team's time when they're drinking coffee half the time and delegating all their work the other half." A sentiment analysis algorithm will see the words "love" and "considerate", positive connotations, in association with "leadership", and will likely classify this comment as positive.

Many are able to negate those words if preceded by "don't" love or "not" considerate, but without those negators, and without any words that are negatively connotated, this comment will be incorrectly classified (Wang, 2017). However, on the whole, it is safe to assume that the majority of comments will not be sarcastic, and insights may be drawn about a comment database as a whole as long as analysts are aware of these limitations. For example, this example above will still teach a model to associate "leadership" with "coffee" and "delegate", which are valuable insights to be aware of.

There is also research which shows that, in survey data specifically, those who bother to comment and write answers to free-response questions are usually a bit more disgruntled, while those satisfied with their workplace environment may leave those optional questions blank (Luo et al., 2016). However, there is also research which shows otherwise (Jipa, 2019). Either way, this is something to keep in mind, that any results drawn from linguistic analysis of survey data come only from the population of respondents who had something to say, not from all respondents as a whole.

Therefore, analysts should find the percent of respondents who answered free-response questions, and use that grain of salt when generalizing results across the population of survey respondents, or the population of the workplace, as a whole. Even if comments trend negatively, additional context provided by the quantitative questions in the survey may indicate that employees are generally content. So, responders' bias should be accounted for.

With respect to topic modeling, there are fewer notorious limitations, but as with all clustering and classification techniques, there are likely to be comments or text that could fall into multiple topic bins. One sentence could address both leadership and training, so decisions need to be made about whether to dual-classify that comment; or associate it with whichever topic it more strongly associates with. Depending on the methodology, whether one is using Support Vector Machines, bottom-up or top-down hierarchical clustering methods, or otherwise, the groupings of the comments by topic may look different. Therefore, just like for sentiment analysis techniques, these methods

are less accurate and less applicable for comments at the individual level, and should be restricted to generalizing insights at a higher, more summative level.

The Data

For purposes in this research, the data are well-suited to the considered methods. The data were compiled by the Financial Management (FM) office at Headquarters Air Force Materiel Command (HQ AFMC) at Wright Patterson Air Force Base (WPAFB). The goal of this survey was to assess employee attitudes about each of four themes: Career Planning, Education & Training, Awards & Recognition, and Recommendations.

The questions were distributed to the FM community in February 2020, to over 7,500 individuals across the Air Force, over 3,000 which were in AFMC specifically. It was available for one week. By the end of that period, they had received 937 responses from AFMC and over 2,000 responses from the entire Air Force, a response rate of about 30 percent. The survey contained four open-ended questions posed about each of the four main themes, and respondents had the chance to write, with no word limit, their thoughts, whether that be criticism, NA, or, less frequently, praise.

The survey guidelines were careful to state that the survey responses would remain anonymous, be studied by analysts outside of the organization, and would in no way come back to reflect on them. Therefore, there would not have been any fear of repercussion to skew the survey responses to be dishonest.

The four open-ended questions, answers to which were included in this analysis, are shown in Table 3, in the order in which they appeared in the survey. The career planning question had significantly fewer responses, and higher negativity bias, because

it was a conditional question based on the response to a previous question. Only respondents who answered that they disagreed with "career advancement and promotion opportunities are adequate" were asked the follow-up question, "What recommendations do you have to make the promotion and advancement opportunities within your area of expertise adequate?".

Table 2: Survey Questions

What recommendations do you have to make the promotion and advancement opportunities within your area of expertise adequate?
Please specify the education and training that are needed to be successful in an FM career as an Air Force civilian employee. Consider the education/training you have already completed as well as the education/training you would like to complete.
What can your organization and/or FM leadership do better to recognize/reward employees?
Please describe any additional recommendations you may have to assist FM leadership in the recruitment, retention, development, and awards/recognition efforts of the FM workforce and/or suggestions to help you perform better in your job.

A team of two analysts in the HQ AFMC A4 office were tasked with assessing and analyzing the survey results, specifically, those of the open-ended questions. In doing so, they spent over 150 man hours manually reading through the answers to those four themed questions. Before reading the responses, they generated a list of subtopics that they expected to see in the responses to each of the four questions, and added some subtopics after reading each comment to create a final list of subtopics addressed throughout. For purposes of linguistics in this paper, a "topic" is one of the four categories posed in the four open-ended questions in the survey. A "theme" is one of a number of sub-topics identified under the umbrella of each of the four topics. For

example, under “Awards & Recognition”, 17 themes were identified, including “Time Off Awards”, “Coins/Certificate/Plaque/etc” and “Letters of Recommendation/Recognition.”

Table 4 indicates the number of responses for each of the four open-ended questions, as well as the average length of response and the standard deviation.

In Appendix B are four tables which state the four main topics, and the themes identified within each, listed in order of prevalence. Prevalence, here, is identified by the number of times the theme is said to have appeared across the survey results.

Table 3: Question Response Metrics

Question	Number of Themes	Number of Answers	Mean Response Length (in words)	Standard Deviation of Response Length
Career	22	200	50.385	41.649
Education	19	675	32.448	29.296
Award	18	612	27.199	29.105
Recommendation	31	483	54.466	61.418
All	90	1970	36.957	43.188

**Mean Response Length and Standard Deviation rounded to 3 decimals

Non-Response Responses

In all, there are nearly 2000 long-form answers in the dataset. The longest answer in the set was 384 words long, in response to the Recommendation theme. Some respondents chose shorter, less meaningful responses, such as N/A, “no comment” and

other similar non-answers. The manual review team designated “NA” or similar as its own subtopic theme for each question.

However, because removing these would require significant manpower to isolate all those responses which, ultimately, provide little information, this research did not remove them all. This is pertinent since one of the goals of this research is to reduce the manpower necessary to accurately draw insights from responses. Only those comments which were some version of “NA” were removed, meaning that comments stating “I have no comment” and “I cannot say” were left in the data. This was identified as the best way forward due to the line being more and more subjective the further one looks at these comments.

For instance, one might disagree that a comment stating “I don’t know” is the same as “NA” since the respondent is indicating a lack of knowledge, rather than a lack of interest in answering the question. This sparks a debate outside the scope of this research. A list of the versions of “NA” present in the comments data is in Appendix A. The non-answer answers removed were all answers to the Education and Training question.

Manual Sentiment Classification

The designated analysts tasked with assessing the results of the survey spent over 150 manhours over the course of two months reading through the responses, not only to document and list those subtopics in each of the four themes, but also to manually document the sentiment associated with that topic in a given response.

For comments on which the two reviewers initially classified comments differently, those comments were re-visited and classified according to a concurrence of discussion between the experts. Then, for each long-form response given in the survey, they have identified up to 7 subtopics, in order of appearance, with associated sentiment classifications for each, and a wider-ranging sentiment classification for the response as a whole. The sentiment classification for the comment as a whole was calculated using the number of themes identified in a given comment and their associated classifications.

For example, a comment with 1 positively-classified theme and 2 negatively-classified themes would be rated negatively overall, whereas a comment with an equal number of negative and positively-classified themes, or a high number of neutrally-classified themes, was rated neutrally. Table 5 summarizes the distribution of the holistic comment classifications across the responses.

Table 4: Manual Sentiment Score Metrics

Question	Number of Negatives (-1)	Number of Neutrals (0)	Number of Positives (1)
Career*	177	23	0
Education	147	456	72
Award	362	151	99
Recommendation	331	132	20
All	1017	762	191

* As explained in the text, this question was conditional and was only posed to respondents who disagreed with the statement, “career advancement and promotion opportunities are adequate.”

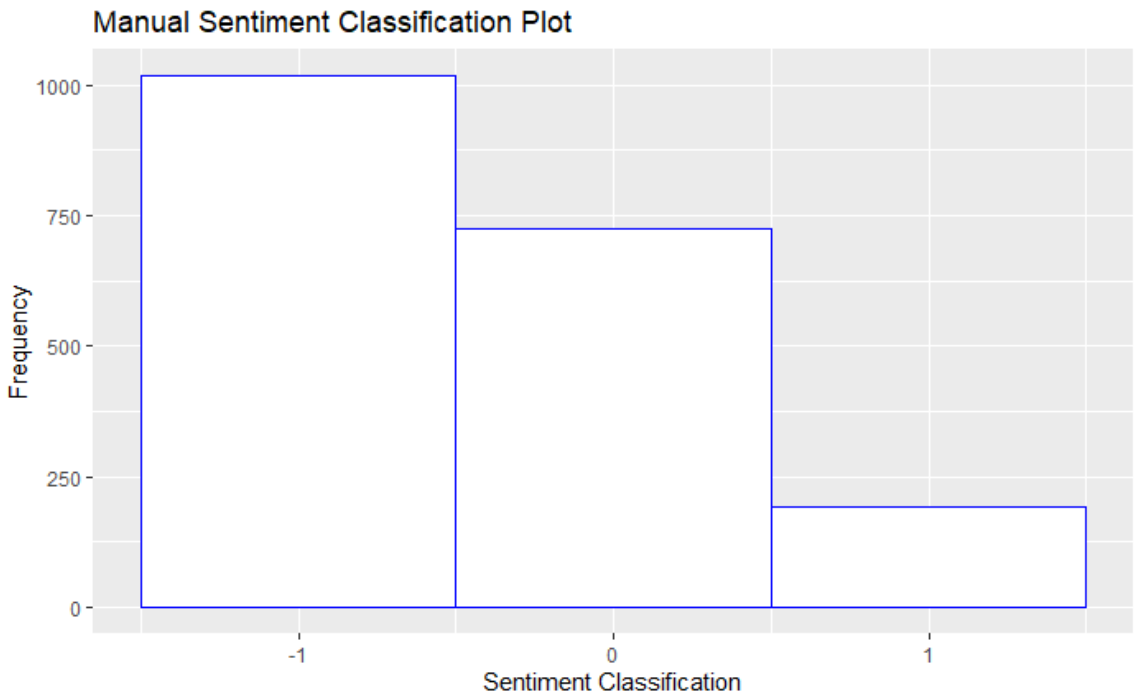

Figure 1: Frequency of Manual Sentiment Classifications

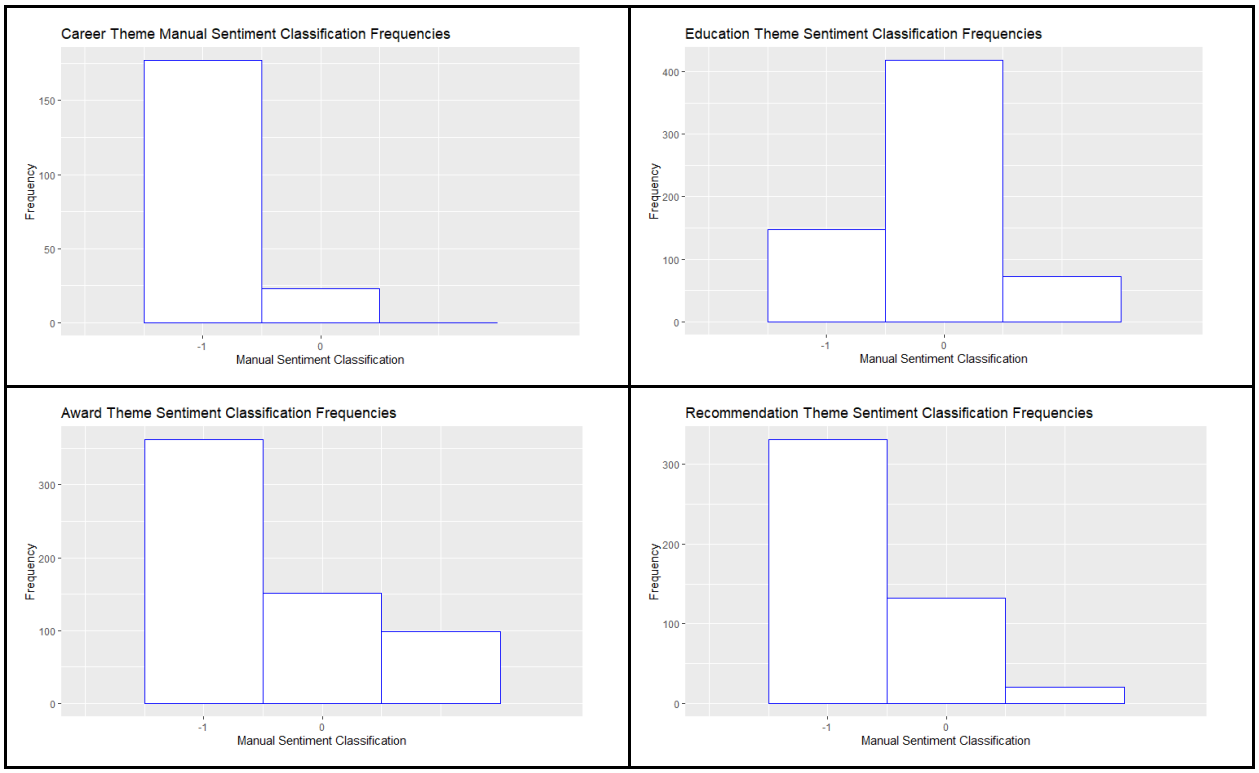

Figure 2: Frequency of Manual Sentiment Classifications by Question

The histograms above reflect the distribution of manually classified scores for the entire set of data and on a by-comment basis.

Sentiment Analysis

The first aspect on which this research focuses is bringing value to the sentiment classifications that the reviewers manually brought to the data. The goal of this portion was to identify the accuracy of existing sentiment analysis methodologies, and to attempt to enhance those that initially performed well, assessing accuracy by association to the manual classifications.

R Programming Language was used for data processing and mathematical manipulation, the code script can be found in Appendix C. In Appendix A is also a list of the comments removed from consideration, those associated with "non-answer" answers.

In this section, the comments were run through various sentiment analysis methodologies, each utilizing different algorithms and drawing from different training sets. Scores were then scaled, and compared against the manual classifications to assess accuracy and variability. These results are reproducible and did not use any degree of randomness.

Table 6 summarizes information about each method utilized in this research for development of sentiment scores.

Table 5: Sentiment Analysis Techniques and Respective Details

Technique	Lexicon options	Published (Naldi, 2019)	Package utilized for method in R	Polarity Scores
BoW with DTM	Syuzhet, AFINN, Bing, NRC	2015	Syuzhet	Weights may reflect intensity of sentiment
BoW with heavy pre-processing	Hu & Liu	2017	Meanr	Weights reflect classification of sentiment
Valence shifters, adversative conjunctions and DTM	Modified combination of Syuzhet and Hu & Liu	2016	sentimentr	Weights may reflect intensity of sentiment
StopWord Removal with heavy pre-processing and BoW Ratio	QDAP, multitude of other options	2017	SentimentAnalysis	Weights may reflect intensity of sentiment

In order to properly compare the performance of each sentiment analysis function, sometimes the same method was with different parameters, and the scores were scaled to mimic those of the manual review classifications. This meant that continuous scores were binned into “positive”, ”neutral” and “negative”.

BoW with DTM

First, the Bag of Words with Document Term Matrix Method was used to calculate polarity scores for each of the comments in the data. This method is largely lexicon-based, a common approach explained above in Literature Review, and gives

users the option to choose the lexicon they want to use. The Syuzhet Package in R was used to implement this methodology.

Essentially, this method takes a bag-of-words approach aided by a document-term frequency matrix. The bag-of-words approach separates the entire document (or, in this case, comment) into a list of words, and then computes a matrix identifying those words that appear next to one another. Without the document term matrix (DTM), there would be no remaining data indicating the structure of the document, for example, if a negator preceded a positive word such as “not happy”.

However, without the additional context that a lexicon provides, the function would not have an idea as to the weight, or perceived negativity or positivity, of a word such as “happy”. This is why there are so many different lexicons available, each developed and trained for different purposes. While the word “happy” is easy to interpret in any context, other words are very context-dependent. For example, the word “faded” may have different connotations depending on if it is describing denim jeans (i.e. positive) or antique furniture (i.e. negative). For each word, depending on the lexicon and the metric used, a polarity is associated, indicating the typical sentiment context in which that word is expressed.

The Syuzhet lexicon was developed by analysts in the Nebraska Literary Lab and ranges from -1 to 1 (Naldi, 2019). The AFINN lexicon began with a set of obscene words developed from Twitter and expanded to include over two-thousand words, including acronyms, and ranges in score from -5 to 5 on a continuous scale (Naldi, 2019). Finally, the Bing lexicon ranges from -1 to 1 and was developed by Minqing Hu and Bing Liu (Naldi, 2019).

Table 6: Information about Applied Lexicons

Lexicon	Number of words	Number of positive words	Number of negative words	Range	Type of Polarity Score
Syuzhet	10748	3587	7161	-1 to 1	continuous
AFINN	2477	878	1598	-5 to 5	discrete
Bing	6789	2006	4783	-1 to 1	binary
Hu & Liu	5787	2005	3782	-1 to 1	binary
Loughran-McDonalds (LM) Financial Dictionary	2709	354	2355	-1 to 1	binary
QDAP	4232	1280	2952	-1 to 1	binary
GI	3642	1637	2005	-1 to 1	binary
HE	190	105	85	-1 to 1	binary

The BoW with DTM Method was applied to the data with respect to three different lexicons, noted in Table 7. These lexicons determined the polarity scores of the words contained within them, and thus had different effects on the sentiment classifications when applied to the data. Summative results of these scores are detailed in the Results section of this paper, along with an assessment of accuracy when compared to the manual sentiment classifications. Because the manual sentiment classifications were discrete, and some of these results are on a continuous or different discrete scale, all results were scaled to match that of the manual results. (Misuraca et al., 2020)

BoW with Pre-processing

This methodology is much simpler than the previous BoW with DTM. Taking in a text string, this method is primarily focused on calculating polarity scores with term-level

polarity aggregations. This is much less advanced than methods previously addressed, and leaves little to no room for customization. This research did not expect this method to perform particularly well in comparison with other, more advanced, methods. However, it was included to test notional assumptions about better methodologies, and this research noted that, when applied to larger datasets, analysts may benefit since the computing time may be significantly faster than other methods due to its simplicity and the ability to utilize parallel computing through the MeanR R Package (Naldi, 2019).

Essentially, taking in a text string, this method includes some pre-processing steps such as punctuation removal and removing capitalization. Then, for each word in the string, if the word appears in the lexicon (in this case, the Hu & Liu lexicon), then its associated polarity is assigned. If the word does not appear in the dictionary, it is assumed that the polarity is zero. Because the Hu & Liu lexicon is discrete, scores for each word are either -1, 0 or 1. Then, the score across the text is computed as the number of positively-scoring words minus the number of negatively-scoring words.

Valence Shifters and Adversative Conjunctions with DTM

This method further builds on and develops some of the concepts mentioned that may improve the performance of a sentiment analysis algorithm. In addition to taking into account negators and amplifiers, it creates a new classification of words encompassing those considered “valence shifters”. These are words that affect the degree to which a word is emphasized or de-emphasized by the writer. It also takes into account “adversative conjunctions”. Therefore, the phrase “very happy” will receive a more

positive score, and “not happy” a more negative score, than the word “happy” on its own would have obtained.

Even when applied to the same lexicons as previous methods the polarity scores will not necessarily be the same. While the BoW with DTM method, for example, would recognize that the words are next to one another, rather than realizing one emphasizes or describes the other, it would treat both words individually according to their polarity score in the lexicon. In this method, rather than “not” and “happy” being treated individually, “not” is instead used to modify, or in this case reverse, the intensity of the polarity of “happy”.

In order to implement this method, the sentimentr package was used (Naldi, 2019). It reads in strings of text as character vectors, and uses punctuation characters to split the string into sentences. The analyst can specify the range that a valence shifter is able to affect. A range of 4 means that in the phrase “not happy, satisfied, or fulfilled”, “not” would be able to affect all three of the adjectives that follow it and shift those polarities, out to 4 words before or after. A range of 1 means that only the polarity of the word “happy” would be affected by the presence of “not”.

Due to the inclusion of malleable valence shifters dictionaries, this method is able to calculate the polarity of text strings not by summing term-level polarity scores or taking the ratio, but considering the words in the context in which they are present. For this reason, the study expected this method to perform better than those previously discussed.

The scoring methodology in the Valence Shifters and Adversative Conjunctions Method computes scores on a by-sentence basis. Therefore, to generate scores of a group

of sentences, the individual scores are averaged and weighted by the word count in each sentence. This risks neutral sentences down-weighting a piece of text, i.e. pulling positive sentence polarity scores down and negative polarity scores up. However, removing the neutral-scoring sentences would disrupt the continuity the study hopes to achieve by comparing metrics across methods, and so the averaging function was left as-is.

Therefore, to calculate comment-level polarity scores with this method, this research used simple weighted averaging (Fuchs, 2020; Raja, 2017).

Stopword Removal with Pre-processing and BoW

The Stopword Removal with Pre-processing and BoW method was more recently developed, and was introduced as a concise SentimentAnalysis R Package in 2019 (Naldi, 2019). It also has the ability to draw from many more lexicons than previous methods addressed in this research. This is useful not only because of the sheer number of lexicons available, but also because of their contextual diversity. The lexicons available for this method include the Loughran-McDonald's Financial Dictionary developed in 2011 and the Qualitative Data Analysis Program (QDAP) dictionaries developed in 2019 (Naldi, 2019).

Unlike some methods previously discussed, this one does not generate scores as the algebraic sum of polarity scores per word or term. Instead, the default score is a ratio of the positive and negative terms. However, this can be changed depending on analyst preferences.

For this research, several methods were used to identify statistically significant words, in part to see the degree of variability between them and assess whether they

indicated any useful insights as to topic-level respondent opinions. Additionally, if any words are context-dependent in the Air Force data and seem to be misinterpreted, and they appear in a list of statistically significant words, that would indicate that the accuracy of the function is way off and could be improved if the polarity of that word, or a set of related words, is adjusted.

Pronoun Adjustment

This hypothesis will be tested by way of the research as explained in a study titled "Employee Pronoun Use In Verbatim Comments As A Predictor Of Job Attitudes And Turnover Intentions", published in 2014 through Wayne State University (Sund, 2017). For each comment, this research will count a total number of "we" and "non-we" pronouns, and calculate the percentage that this accounts for in the total words used. Appendix A of this paper details the pronouns in each category, and can be reproduced.

In organizational psychology, this is called relationship literature, and pulls from the notion that pronouns of the "we" type indicate that the writer feels a sense of unity and community with their workforce and peers, while the use of "non-we" pronouns may indicate distancing and dissatisfaction between the author and their workforce (Slatcher & Vazire, 2008). For this research, a correlation matrix will then be created as demonstrated in the previously referenced research, and an ANOVA table will be used to assess the degree to which sentiment scores are correlated with pronoun usage.

Because the polarity of pronouns were said to influence the study, the first attempt at improving the performance of the SentimentAnalysis package was to adjust the stopword removal in the pre-processing phase to allow certain pronouns to remain in the

document term matrix. Then, the SentimentLM lexicon, which did not currently contain any of these pronouns, was amended to include them, with either strictly negative or strictly positive weights associated with them since the SentimentLM lexicon is a binary dictionary. Several combinations of including pronouns in the positive and negative dictionaries were attempted, and the best combination for improving overall accuracy seemed to be weighting "you" and "they" pronouns negatively.

Context Word Adjustment

A similar approach was attempted to improve the performance of the dictionary with respect to Air Force specific words. While one may notionally be able to identify words that they may assume are context specific, if the word does not have a polarity at all, it will not be swinging the scores in the wrong direction, rather, it just will not contribute. However, words that are incorrectly classified may have a much bigger impact on the model performing poorly.

Therefore, to identify those at-risk words, the LASSO method was used in conjunction with regression analysis to identify words that contribute more heavily to the scores in the model. LASSO stands for least absolute shrinkage and selection operator), a regression analysis method originally formulated for application to linear regression models in attempts to improve prediction accuracies and model interpretability. Ordinary least squares (OLS) or generalized linear models (GLM) could have also been used.

Table 8 shows the list of words generated when compared to a variety of scores. This was not only run with the Manual, accurate scores, since for an analyst to identify at-risk words, they may not always have access to those manual scores. So, the study

wanted to identify whether similar words appeared with other scoring mechanisms. Words have been stemmed by the pre-processing step, which is why some may look different than the direct terms. The four methods that had the highest strict accuracy scores thus far were investigated in this manner.

Table 7: Words identified by LASSO method as statistically significant, by scores

Manual scores	BoW with DTM (Bing Lexicon)	Valence Shifters and Adversative Conjunctions w DTM	Stopword Removal and Pre-processing with BoW Ratio (Sentiment HW Lexicon)	Stopword Removal and Pre-processing with BoW Ratio (Sentiment LM Lexicon)	Stopword Removal and Pre-processing with BoW Ratio (Sentiment HW Lexicon) with Pronoun Adjustment
Intercept: -0.3487515 -0.04 peopl -0.03 posit -0.03 get -0.02 award -0.02 employe -0.02 work -0.01 need -0.01 opportun 0.01 job 0.02 train	Intercept: 0.5857396 0.01 train 0.03 get 0.03 level 0.06 time 0.07 leadership 0.07 peopl 0.10 employe 0.11 job 0.12 need 0.13 opportun 0.22 work 0.26 award	Intercept: 0.1696322 -0.01 get -0.01 leadership -0.01 peopl 0.01 level 0.01 award 0.04 opportun	Intercept: 0.04444311 0.01 posit 0.02 opportun	Intercept: 0.0430529 -0.01 need 0.01 posit 0.01 opportun 0.02 leadership	Intercept: -0.001219411 -0.20 get -0.16 job -0.12 posit -0.10 work -0.09 peopl -0.05 need -0.04 time -0.03 train -0.02 award -0.02 employe -0.01 level 0.05 opportun 0.22 leadership

As one looks across Table 8, note that outside of the first column, the methods are ordered by total accuracy percentage. One can see that the words identified as statistically significant in their contribution to the scores is similar across the different columns of the table. Statistically significant words with coefficients effectively at zero were not

included in this table. These were tested at a 0.05 significance level. Coefficients which are negative contributed to a negative weight when computing the sentiment scores, whereas coefficients which are positive contributed to a positive weight when computing the sentiment scores.

However, some words have negative coefficients in their contribution to the score, and have positive coefficients in their contribution to a different score. “Peopl” is always negative and shows up in each column. The word “get” is in each column, but is sometimes positive and sometimes negative. Words with a wider span and larger coefficient likely contribute more to the variability between the models. However, none of these seem to be largely context-dependent.

Regardless, the study found that in the SentimentLM dictionary, the words “posit” and “opportun” were in the dictionary as positive words. Therefore, an attempt was made to remove those words from the dictionary, since they are subjects of the question and thus should not have a polarity associated with them, and a new accuracy score was calculated. The results of that attempt are explained in Conclusions.

Topic Modeling with LDA

The second aspect of this research was to assess the use of Topic Modeling methods to identify key themes in the data, sorted by Question. The manual review team went into the data having already defined a list of topics that they expected to see, but adjusted that list after having read through the comments and seen the trends.

In order to prepare the data for topic modeling, it was first run through several pre-processing steps. All capitalizations were switched to lowercase, stopwords were

removed, punctuation was removed, numbers were removed, whitespace was removed, and all words were stemmed. Then, a document term matrix was created with a minimum frequency parameter set to 5, meaning words that appear fewer than 5 total times throughout the data will not be in the DTM. Then, empty rows of the DTM were removed, and the data was ready for LDA - based topic modeling.

The function LDA() from the R Package topicmodels was used to conduct this step (Jagtap et. Al., 2013). This function asks the user for the expected number of distinct topics, a method to be used for fitting, and other parameters such as the desired number of iterations. For the first pass in this research, the function was run over all of the response data, in order to determine if the model would be able to separate the topics of the four main questions. For this reason, k=4. The Gibbs Method was used for the fitting method, and 500 iterations were conducted with the verbose parameter set to 25. However, when calculating the ideal number of topics k when doing topic modeling, there are many methods for choosing the ideal k value (Schweinberger, 2020).

The top 10 terms for each of the 4 topics identified are displayed in the Conclusions. The model was also run with k=3 since the final question was very broad and respondents could have answered with comments pertaining to any of the previous three. For this iteration, responses to the fourth question were removed, in order to isolate the performance of the model on those responses to the three clear topic-oriented questions.

IV. Analysis and Results

Chapter Overview

In the subsequent tables below is information reflecting the results of the applied methods with the comments in the data. Comparison between each method's performances can be found in Table 11.

Queries about the code behind these methods can be found in the respective package libraries, which explain in detail the functions contained within packages and the arguments that the use may pass to those functions. Additionally, the code written for this research is available in Appendix C.

BoW with DTM Method Results

This method was applied with four different lexicons: Syuzhet, Bing, AFINN and NRC. As indicated in Table 9, the Syuzhet polarities ranged from -3.25 to 15.90, the Bing polarities ranged from -7 to 12, the AFINN polarities ranged from -13 to 36, and the NRC polarities ranged from -6 to 17. Table 9 indicates the range of the polarity scores at the comment level, and the frequency per bin when scaled for comparison with the manual scores.

Figure 6 indicates the distribution of the scores for each of the methods. These show that Bing was more centered on 0 while Syuzhet and NRC had longer tails into the positive scores. The distribution of the discrete scores from the manual reviewers is shown in Table 9 and Figure 6, with the number of negative scores being much higher than the number of positive scores and the number of neutral scores. In scaling the scores

obtained here, scores greater than 0 are classified as positive, less than zero as negative, and zero as neutral. These are revisited for comparison in the Summaries section.

Table 8: BoW with DTM Method Lexicon Results

Lexicon	Min Polarity	Max Polarity	Average
Syuzhet	-3.25	15.90	1.69
Bing	-7.00	12.00	1.00
AFINN	-13.00	36.00	3.04
NRC	-6.00	17.00	2.05

*Polarity scores rounded to 2 decimal places

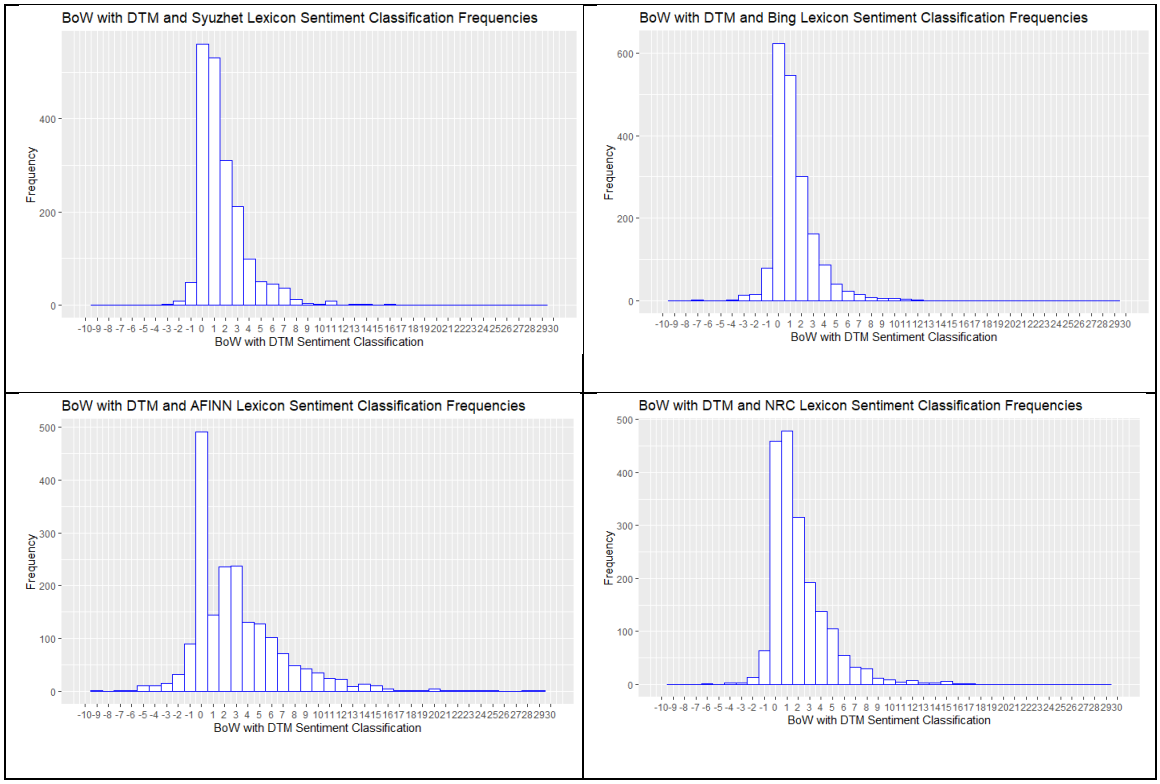

Figure 3: BoW with DTM Method Frequencies by Lexicon

Pre-processing with BoW Method Results

Pre-processing with BoW was applied to just one lexicon, the Hu & Liu. As indicated in Table 10, the polarities ranged from -4.00 to 19.00. Figure 4 indicates the distribution of the scores. These show that sentiment was somewhat centered on 0 with a long positive tail. In scaling the scores obtained here, scores greater than 0 are classified as positive, less than zero as negative, and zero as neutral. These are revisited for comparison in the Summaries section.

Table 9: Pre-processing with BoW Method Results

Lexicon	Min Polarity	Max Polarity	Average
Syuzhet	-4.00	19.00	1.51

*Polarity scores rounded to 2 decimal places

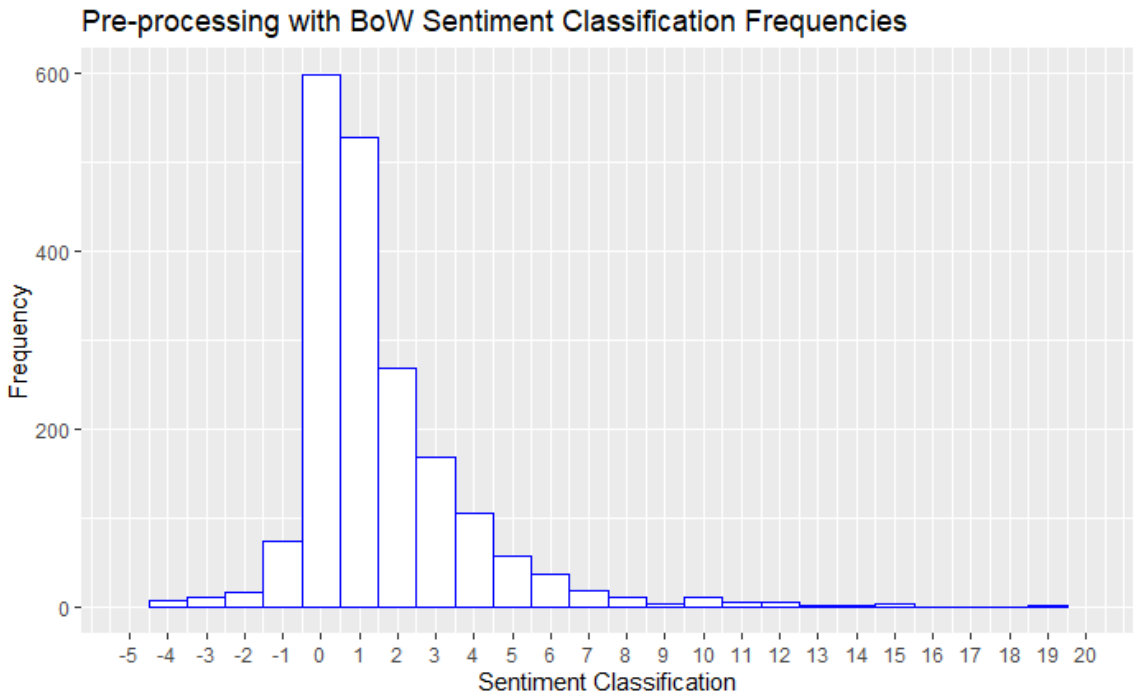

Figure 4: Pre-processing with BoW Method Frequency Graph

Valence Shifters and Adversative Conjunctions Method Results

The Valence Shifters and Adversative Conjunctions Method was run over the vector of comments from the original data. There are options in the dynamic parameters to customize the valence shifter dictionary and the number of terms surrounding the valence shifter that it may affect.

The analyst may also alter the dictionary, as well as the weights of the valence shifters. As noted in methodology, the function was run once with the downweighted averaging function, and once with the average mean.

As one can see in Table 10 and Figure 5, the averaging method did not make a significant difference when compared to the down-weighted zeros, and so only scores from the first row method will be used for comparison.

Table 10: Valence Shifters and Adversative Conjunctions Method Results

Averaging Method	Min Polarity	Max Polarity	Average Polarity Score
average_mean	-1.125	2.221	0.1839
Default: down weighted zeros	-1.125	2.221	0.1816

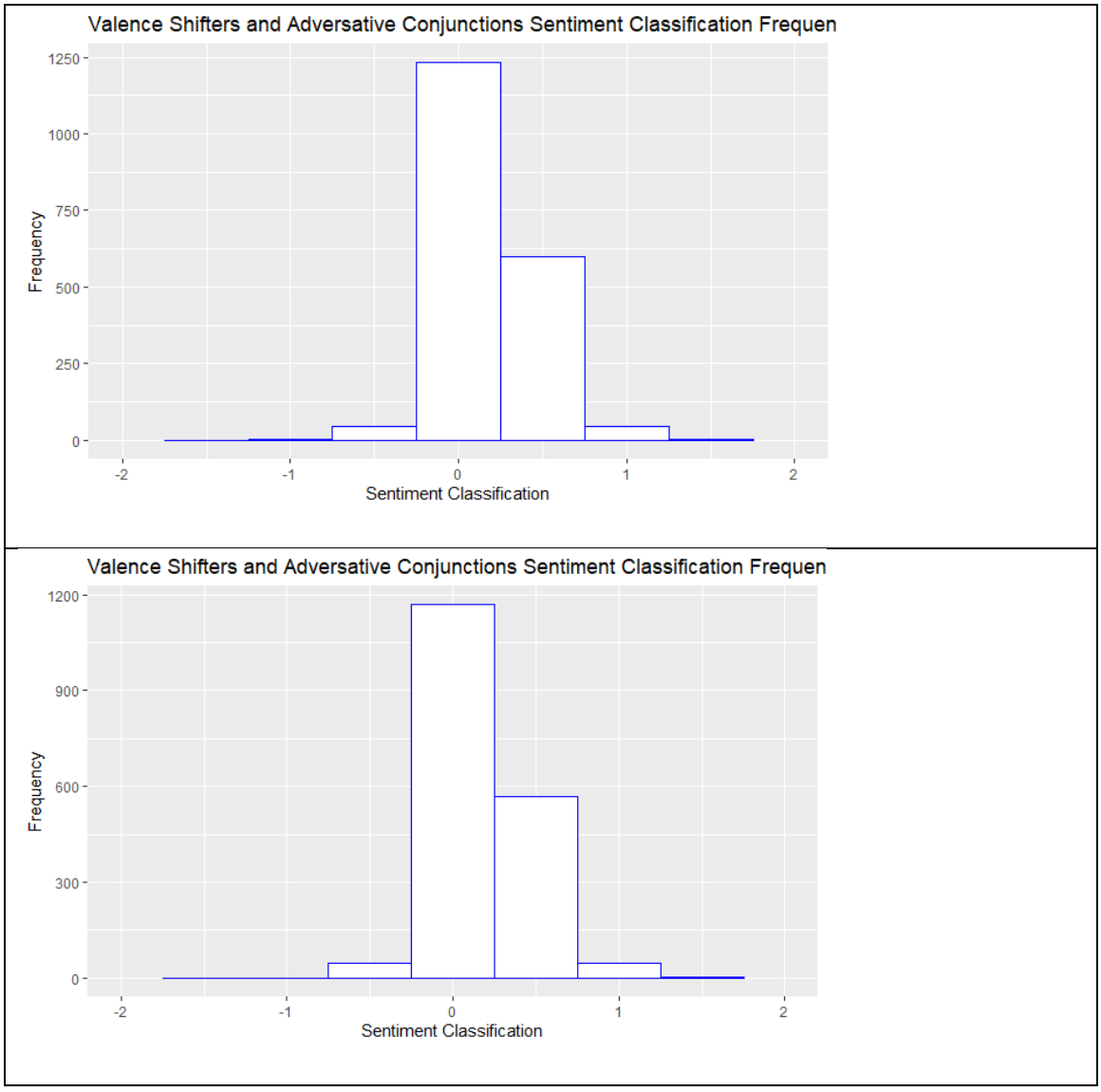

Figure 5: Valence Shifters and Adversative Conjunction Results Frequency

Comparison Results

Figure 6 shows correlation matrices indicating the results of the initial sentiment analysis methods. The correlation indicates similarity or dissimilarity of word polarity between method and manual scores.

Manual scores are not strongly associated with any of the automated methods, with the highest absolute correlation (via the Pearson statistic) being 0.177 with the valence shifters approach. However, several of the automated methods are highly correlated. For example, the BoW with DTM Method, with Bing lexicon, and the Pre-processed BoW method, are very highly correlated, with r = 0.95. Figure 6 shows correlation of scores when binned for direct comparison to the manual scores

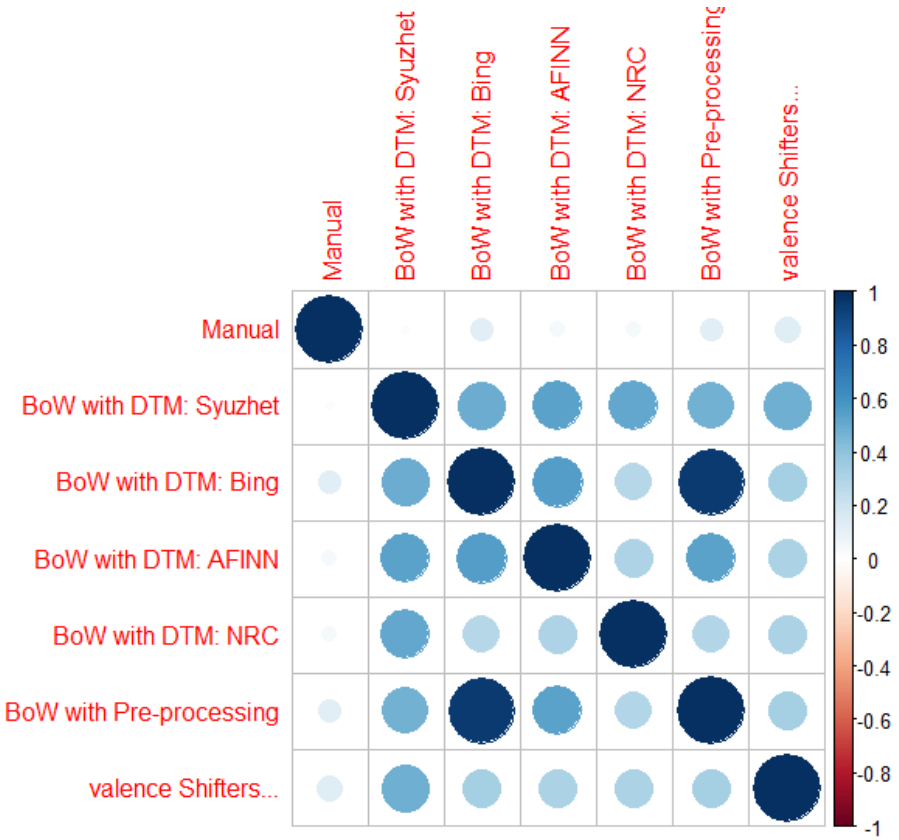

Figure 6: Correlation Table of Results from Automatic Methods

Figure 6 and Table 11 indicates associations between the methods tested thus far beyond correlation, investigating degrees of accuracy. Equations for each of the columns are explained below.

Total Accuracy is the sum of the correctly classified positive comments, the correctly classified negative comments, and the correctly classified neutral comments, divided by the total number of comments being classified.

Total Accuracy = (# correct Positives + # correct negatives + # correct neutrals) / total

Positive Accuracy is the sum of the correctly classified positive comments divided by the total number of manually-classified positive comments.

Positive Accuracy = (# correct Positives) / total True Positives

Negative Accuracy is the sum of the correctly classified negative comments divided by the total number of manually-classified negative comments.

Negative Accuracy = (# correct Negatives) / total True Negatives

No Neutral Accuracy is an attempt to remove from consideration those comments whose aggregate manual sentiment classification may be incorrect. Therefore, this calculates the performance of the algorithm when the manually-classified neutral comments are removed, and it is the sum of the correctly classified positive and negative comments, divided by those comments which were manually assigned a sentiment other than neutral.

NoNeutral Accuracy = (#correct Positives + #correct Negatives) / (total - True Neutrals)

However, the Total Accuracy percentage may be perceived to be skewed due to the way in which the manual sentiment scores were classified for comments that expressed multiple sentiments. When those 97 comments manually classified as neutral

are removed, the accuracy percentages shift, and the distribution of the classifications are instead 984 negative, 718 neutral and 171 positive. However, the manual scores became less correlated with the computer-generated scores, and the accuracy results did not significantly improve.

This research did not choose to remove the manually-classified neutral comments from consideration since it would not be realistic for a team of analysts to remove all those comments containing multiple sentiments from consideration without first having read through them.

One Sentiment Accuracy is calculated as the total accuracy for comments that only expressed one sentiment according to the manual review team. Therefore, comments that expressed multiple sentiments, and were thus averaged for the total manual score, were removed. This is to investigate the degree to which that manual sentiment method affects the accuracy scores. These are computed using the scaled results.

Table 11 indicates that the algorithms are very good at correctly classifying negative comments, but have low accuracy classifying positive comments. The effect is to drive down the accuracy of the package as a whole (Silge & Robinson, 2021)

Table 11: Method Accuracy Results

Method	Total Accuracy	Positive Accuracy	Negative Accuracy	No Neutral Accuracy	OneSentiment Accuracy
BoW with DTM (Syuzhet)	0.227	0.069	0.948	0.208	0.226
BoW with DTM (Bing)	0.306	0.094	0.906	0.223	0.306
BoW with DTM (AFINN)	0.302	0.116	0.890	0.238	0.304
BoW with DTM (NRC)	0.232	0.057	0.869	0.185	0.229
Pre-processing with BoW	0.295	0.090	0.906	0.219	0.295
Valence Shifters and Adversative Conjunctions with DTM	0.307	0.225	0.942	0.339	0.308

**all scores rounded to 3 decimal places

Stopword Removal with Pre-processing and BoW Method Results

Up to this point, the method performing best accounted for valence shifters and adversative conjunctions, which had an accuracy of 30.7% with the manual scores. Stopword removal with pre-processing and BoW were adjustments added in an attempt to improve accuracy. Table 12 shows the results from one run of the method, drawing from 4 different lexicons (Feuerriegel & Proellochs, 2019; Misuraca et al., 2020).

Table 12: Lexicons for Valence Shifters and Adversative Conjunctions Method

Lexicon	Total Accuracy	Positive Accuracy	Negative Accuracy	No Neutral Accuracy
SentimentGI	0.217	0.080	0.911	0.211
SentimentHE	0.308	0.036	0.534	0.115
SentimentLM	0.374	0.207	0.675	0.281
SentimentQDAP	0.236	0.050	0.932	0.190

**all scores rounded to 3 decimal places

Table 12 shows this enhanced method performs better in total accuracy and better in correctly classifying positive comments when compared to the previous methods. However, the accuracy for correctly classifying negative comments is significantly lower.

Table 13 shows the results of the Pronoun improvement attempts on this method with the SentimentLM lexicon as detailed in Methodology.

Table 13: Accuracy Results with Pronoun Adjustment

Method	Total Accuracy	Positive Accuracy	Negative Accuracy	No Neutral Accuracy
SentimentLM	0.374	0.207	0.675	0.281
You added to Negative	0.451	0.694	0.182	0.435
You and They added to Negative	0.471	0.678	0.180	0.479
You and I added to Negative	0.454	0.688	0.180	0.442
You: Negative We: Positive	0.451	0.693	0.182	0.435
You, They and I Negative	0.446	0.694	0.174	0.414

An additional attempt to improve the performance of this method was to remove those words that were subjects of the question posed to respondents and listed as significant contributors to the score as a whole, and in the lexicon from which the function was pulled. The result was the removal of two stemmed wordsm “posit” and “opportun”. After removal from the dictionary, the scores were recalculated and the accuracy re-assessed. Table 14 displays the results. Even in conjunction with the pronoun rule previously implemented, and currently with the highest total accuracy, this did not improve that approach to any significant degree.

Table 14: Accuracy Results with Context Word Adjustment

	Total Accuracy	Positive Accuracy	Negative Accuracy	No Neutral Accuracy
Words pulled from SentimentLM	0.414	0.712	0.176	0.363
Words pulled from SentimentLM & Pronoun Approach	0.471	0.678	0.180	0.479

Ultimately, the best-performing model was the Valence Shifters and Adversative Conjunctions Method with the SentimentLM lexicon, amended with the pronouns dictionary. Table 15 summarizes the results of the comparison between method scores and the manual assessment.

Topic Modeling with LDA

Table 15 shows the 10 top terms, by frequency, in the 4 themes and 3 themes, respectively, identified by application of Latent Dirichlet Allocation.

Table 15: LDA topic modeling results

K=3		
Topic 1 terms	Topic 2 terms	Topic 3 terms
Train Level Career Requir Degr Cour Manag Certif Educ Field	Employ Award Work Time Peopl Leadership Supervisor Program Make recogn	Job Posit Promot Opportun Perform Process Experi Organ Provid Year

V. Conclusions and Recommendations

Implications

All results drawn from this analysis are only directly applicable to the specific context of the survey data on which the methodologies and models were tested. That survey data is centric to the Financial Management civilian career field in the United States Air Force.

This research indicates that automated sentiment classification techniques are insufficient in garnering the sentiment of a piece of text as a whole when applied as the sole classification method. Instead, analysts should pair sentiment classification techniques with careful parameters which better suit the algorithm to the context to which it is being applied. This may mean implementing one or more of the adjustment techniques explored in this research, building a lexicon for the analysis use-case specifically, or training an algorithm on a smaller set of manually classified, taken-as-truth, classifications to the set. In this research, the truth data metric, i.e. the manually assigned classifications, may not have been nuanced enough to allow for sufficient comparison to the automated techniques. Manual classifications calculated in a different manner may yield different results.

Additionally, the topic modeling algorithms seemed promising in their application to the data, when the attempt was made to use that algorithm to sort answers to all of the questions into the main themes addressed. Therefore, it may be a viable recommendation that analysts first identify the topics which they wish to further investigate in a dataset of textual comments, and then apply sentiment analysis as a secondary technique, thereby

parsing the data into a smaller set and isolating those comments which address a given topic. If techniques are applied in this order, not only can the analyst then apply sentiment analysis techniques specifically to a unique lexicon, whose use-case is clear, but it may also be more viable to visually check the performance of a sentiment analysis classification against those comments to which it is being applied.

Any assumptions about degrees of confidence in the tangential application of these techniques to other United States Air Force career fields, other organizations in the government or Department of Defense, or parallel career fields for active duty and enlisted employees, should be carefully verified. While this research is intended to bring a greater degree of understanding to linguistic analysis in an Air Force context, implications about expected degrees of accuracy should be verified when applied to new contexts.

Recommendations for Future Research

Future research could include further exploration of some of the techniques identified as more effective, with higher accuracy. Analysts may also explore the degree to which a lexicon developed specifically for application to a Department of Defense or Air Force specific context may perform when compared with generalized lexicons. More study is needed to identify the aspects of this field of study which influence accuracy percentages and the performance of sentiment analysis models. Aspect-level sentiment analysis, even when paired with flawless topic modeling, would not perform well given the lack of confidence in sentiment analysis results, even at the sentence level, for either specific identification or generation of summative numbers.

Appendix A: Non-Response Responses and Pronouns

Non-response Responses: ("na","NA","na/","N/A","n/a")

Table 16: Pronoun Lists

Pronoun Category	Words Included
We	we, our, ours, ourselves, we're, we've, we'd we'll
I	I, I'd, I'll, I'm, I've, me, my, myself
You	you, you'd, you'll, you're, you've, your, yours, yourself, yourselves
They	they, they'd, they'll, they're, they've

Appendix B: Subtopics Identified By Theme

Table 17: Career Themes

Theme	Number	Count	Theme%
Perceived favoritism/promoting friends/already know who/need merit-based	2	37	12.94%
Lack of Lead/Non-supervisory/Specialists Positions	18	35	12.24%
Education/PME over practical experience when promoting (box checking)	11	33	11.54%
AF FM positions graded lower than other DoD agencies/career fields or fewer high grade	12	32	11.19%
Poor Leadership/Communication	15	27	9.44%
No Career Paths/Limited Opportunities for Growth	17	27	9.44%
Hiring and Interview Process Issues (e.g., certs)	19	27	9.44%
Lack/insufficient Mentoring	5	10	3.50%
Development training opportunities	10	9	3.15%
Pay band vs GS	13	9	3.15%
Cross-functional career broadening opportunities	4	7	2.45%
Hiring/promoting from within underused/DHA	6	6	2.10%
Outdated PDs not matching actual workload	1	5	1.75%
Internal rotations to gain more breadth (eg, 1515	8	5	1.75%
N/A	22	5	1.75%
Training plans	7	4	1.40%
Better OJT	16	3	1.05%
Organizational Structure	20	2	0.70%
Toxic work environment	9	1	0.35%
Incentives for hard-to-fill positions (ie, 1515)	14	1	0.35%
Overworked Supervisor	21	1	0.35%
More target developmental positions	3	0	0.00%

Table 18: Education Themes

Theme	Number	Count	Theme%
Training OJT (Systems and Tools, Onboarding into new job, job-specific Policy, DTS, GPC, EVM, local training, etc.)	11	188	15.64%
Certification req (FM, APDP-DAU)	2	161	13.39%
Education (BA, MA, PhD, AFIT, Naval post grad)	8	118	9.82%
Training Functional/Technical (Acct, Audit, Budget, Cost, Finance, FM Bootcamp, FM basics, Fiscal Law, general policy, etc.)	13	108	8.99%
PME	4	106	8.82%
CDE	5	104	8.65%
Training Non-Functional/Technical (Critical thinking, Analytical, Communication Skills, leadership, managing people/generational gap, time mgmt, data analytics, programming, focus week, MS Office, EVM, FMS, etc.)	14	95	7.90%
Barriers to training (Workload/Timing/Location and Travel~Work Life/Limited Seats/Not selected/Unable to rotate/job series makes inelligible)	16	50	4.16%
None/Too much already/Training before experience/don't need any	17	41	3.41%
N/A or Don't know or Not participated or No comment or unsure	18	36	3.00%
Not req (CDFM, CPA, unspecified, AFIT Certs like Data Analytics, Prof Orgs etc.)	1	34	2.83%
Leadership (Perceived favoritism, leaders don't fight for their people, comm/awareness, etc.)	15	29	2.41%
Training for new to Gov/Trainee/PAQ	10	25	2.08%
Job Experience Cross Functional (Career Broadening, Rotations, Multi orgs)	12	24	2.00%
Box checking/Promotion/hiring/Unequal consideration of training required	19	23	1.91%
Job Experience Gov/Mil/Industry	3	19	1.58%
Mentoring/Supervisor/Peer Guidance	7	17	1.41%
Any training is good	9	16	1.33%
Conferences/Workshops/PDI	6	8	0.67%

Table 19: Award Themes

Theme	Numb	Count	Theme
Utilization of Awards Program (Leadership usage/lack of, "putting in the time to do it", "its their turn/spread the wealth")/Consistency/some supervisors or orgs nominate more	9	171	20.53%
Bad work environment/leadership <supervisors need time/skill/awareness of who does what or too dependent on their time/skill/awareness AND get out and see people/motivate and push folks to do more?/know what the employee wants><mentorship, lame ducks>	15	166	19.93%
Perceived Bias in favorites/age/etc. or Equalize Supervisor use (utilization maybe) or always same people or overfocused $s <how to balance (big picture vs daily, high vis and workload vs not, big vs small programs, MAJCOM vs Base, provide award opps, move people around) [utilization]><merit/hardwork,FM vs. FMS/1515/NAF>	13	81	9.72%
Awards Program/Process (what it takes to nominate, ability self-nominate and self promote, time to do it right, timeliness, timelines, etc.)/Consistency/train people on it/standardize it/writing exercise	10	75	9.00%
Verbal Recognition/Thank you's/On the spot/informal "pat on the back"/Let them know how they have contributed to the larger mission [THANK YOU]	7	57	6.84%
Time Off Awards	1	55	6.60%
Other Awards and Recommendations/Not so individual focused	4	50	6.00%
Money/Increase in Pay/Promotion	2	48	5.76%
N/A or Don't know or None received or No comment or unsure	18	29	3.48%
Communication/Awareness of awards being given and what are available/who was nomin.	11	28	3.36%
FM vs. Other Functionals (utilization or bias maybe)/AF vs Other Gov	12	26	3.12%
Public Recognition (director's calls to group meetings)/face time with leadership	8	16	1.92%
Coins/Certificate/Plaque/etc.	3	7	0.84%
Flexible work schedules/telework/CWS/~when and how we work/awards year round/wingman days and events	14	7	0.84%
Team Awards Competitive or otherwise (not duplicative of above)	5	6	0.72%
Letters of Recommendation/Recognition	6	5	0.60%
No Awards Are Necessary	17	4	0.48%
No Preference/Any/All are good/Doing a good job	16	2	0.24%

Table 20: Recommendation Themes

Theme	Numbe	Count	Theme%
Leadership/Work Environment/Supervisor Training/Lack of Appreciation/Get to know folks and understand	5	113	13.61%
General Recommendations (non of the above/new ideas/things we don't do)	26	70	8.43%
Hiring and Interview Process (e.g., certs), DHA?	7	58	6.99%
Flexibility (Telework, flex work schedules, CWS)/PT/Mobilityish	16	53	6.39%
No Career Paths/Limited Opportunities for Growth (Broadening, Good Projects)/Bad promotion process/No Succession Plans/PAQs get more than new to gov/Opportunities for career development (Roadmap, Rotations, Broadening, cross functional,Good Projects)	8	51	6.14%
Training OJT (Systems and Tools, Onboarding into new job, job-specific Policy, DTS, GPC, EVM, local training, etc.)	9	49	5.90%
Perceived bias favoritism/promoting, hiring, training selection, or awarding friends/favorites	1	40	4.82%
Utilization of Awards Program (Leadership usage/lack of, "putting in the time to do it", "its their turn/spread the wealth")/Consistency/some supervisors or orgs nominate more	11	39	4.70%
Pay/Compensation/Benefits/Pay Equality (sup vs non sup)	29	31	3.73%
Education/PME over practical experience when promoting (box checking)	3	30	3.61%
Communication/Transparency	6	30	3.61%
Pay Systems (AcqDemo vs LabDemo vs GS, everyone needs to be on the same thing, etc.)	22	21	2.53%
Training Functional/Technical and Non-Functional/Formal Education/PME/CDE	10	20	2.41%
Non/Low-performers (includes trainees)/Merit & work based/hold accountable	17	19	2.29%
Trainee program for all (PAQ, Trainees, New to Gov)	30	19	2.29%
Mentoring	14	18	2.17%
Funding for training, education, etc. (Tuition Assistance)	24	18	2.17%
Awards Program/Process (what it takes to nominate, ability self-nominate and self promote, time to do it right, timeliness, timelines, etc.)/Consistency/train people on it/standardize it/writing exercise/NEED TRAINING	12	17	2.05%
AF FM vs. Not & Acq vs. Non-Acq vs. Other functionals/agencies (position grading)	4	16	1.93%
Verbal Recognition/Thank you's/On the spot/informal "pat on the back"/Let them know how they have contributed	13	16	1.93%
Overworked/workload/Workload is a barrier to other things such as training/mentoring/balance workload	31	16	1.93%
Facilities/IT infrastructure	18	13	1.57%
Awards TOA/Monetary/Salary	28	13	1.57%
Workforce balance (Internal vs External hiring, balance of trainee/PAQ/CTR)	21	12	1.45%
N/A	27	10	1.20%
Culture (Beauracratic, stuck in our ways, slow to react)	20	9	1.08%
Utilization of workforce/Correct PDs	15	8	0.96%
FM Systems and LMS	19	8	0.96%
Certification Requirements	23	8	0.96%
Specialists vs Generalists vs Supervisory or Not	2	3	0.36%
Appraisals	25	2	0.24%

Appendix C: R Code Script

````
title: "Thesis.Work"
author: "Julia Haines"
date: "1/16/2021"
output: pdf_document
---
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo =
TRUE)
```
## Importing Libraries
```{r, echo=FALSE,
warning=FALSE, message=FALSE}
options(java.parameters = "-
Xmx8000m")
library(tm)
library(ggplot2)
library(openxlsx)
library(topicmodels)
library(tidyverse)
library(tidytext)
library(slam)
library(stringr)
library(dplyr)
library(tidyr)
library(lsa)
library(LSAfun)
library(ngram)
library(reticulate)
library(sentimentr)
install.packages("sos")
library(sos)
#findFn("str_replace")
py_module_available("gensim")
py_module_available("pyLDAvis")
setwd("C:/Users/hainjm15/Documen
ts/AFIT")
'%ni%' <- Negate('%in%')
```
## Loading the Data with Summary
Statistics
This piece also identifying
nullified responses and creating
column ManualSent for aggregate
classification
```{r, echo=FALSE}
data<-
read.xlsx("AFMC.CommentData.xlsx
",sheet=3)
afdata<-
read.xlsx("AF.CommentData.xlsx")
````
````

```
data1<-data
data3<-data[-
which(data$AvgSent=="Multi"),]
questions<-c("Career
Planning/Succession Planning
27","Education & Training
34","Awards and Recognition
35","Recommendation to Career
Field 36")
#counting the number of
sentences per comment and
finding the mean and standard
deviation for summaritive
purposes
data$length.response<-
sapply(strsplit(data$Response, "
"), length)
sd(data$length.response)
max(data$length.response)
#finding the mean and sd of by-
question sentence length per
comment
mean(data$length.response[which(
data$Question=="Recommendation
to Career Field 36")])
sd(data$length.response[which(da
ta$Question=="Recommendation to
Career Field 36")])
#calculating Manual Sentiment
column
data1$count.pos <- apply(data1,
1, function(x)
length(which(x=="Positive")))
data1$count.neg <- apply(data1,
1, function(x)
length(which(x=="Negative")))
data1$count.neu <- apply(data1,
1, function(x)
length(which(x=="Neutral")))
summary(as.factor(data3$AvgSent)
)
length.1<-dim(data1)[1]
data1$ManualSent<-
rep(0,length.1)
data1$MSNum<-rep(0,length.1)
for (i in 1:length.1){
 if (data1$count.pos[i] >
max(data1$count.neg[i],data1$cou
nt.neu[i])){
   data1$ManualSent[i]<-
"Positive"
   data1$MSNum[i]<-1}
  else if (data1$count.neg[i] >
max(data1$count.pos[i],data1$cou
nt.neu[i])){
```

```
   data1$ManualSent[i]<-
"Negative"
   data1$MSNum[i]<--1
 }else if (data1$count.neu[i] >
max(data1$count.pos[i],data1$cou
nt.neg[i])){
   data1$ManualSent[i]<-
"Neutral"
   data1$MSNum[i]<-0
 }else if
(data1$count.pos[i]==data1$count
.neg[i]){
   data1$ManualSent[i]<-
"Neutral"
   data1$MSNum[i]<-0
 }else if
(data1$count.pos[i]==data1$count
.neu[i]){
   data1$ManualSent[i]<-
"Positive"
   data1$MSNum[i]<-1
 }else if
(data1$count.neg[i]==data1$count
.neu[i]){
   data1$ManualSent[i]<-
"Negative"
   data1$MSNum[i]<--1
 }
}
#converting Manual Sent to
numeric equivalent
#breakdown of sentiment
classifications across comment x
for (i in 1:4){
  quest<-questions[i]
print(summary(as.factor(data1$Ma
nualSent[which(data$Question==qu
est)])))
}
summary(as.factor(data1$ManualSe
nt))
voided<-
c("na","NA","na/","N/A","n/a")
data1<-data1[-
which(data1$Response %in%
voided),]
data1<-
data1[which(!is.na(data1$Respons
e)),]
data3<-data3[-
which(data3$Response %in%
voided),]
data3<-
data3[which(!is.na(data3$Respons
e)),]
hist(data1$MSNum)
```

```
par(mfrow=c(2,2))
hist(data1$MSNum[which(data1$Question=="Career Planning/Succession Planning 27")], main="Q27 Distribution",xlab="polarity",ylab="Freq")
hist(data1$MSNum[which(data1$Question=="Education & Training 34")],main="Q34 Distribution",xlab="polarity",ylab="Freq")
hist(data1$MSNum[which(data1$Question=="Awards and Recognition 35")],main="Q35 Distribution",xlab="polarity",ylab="Freq")
hist(data1$MSNum[which(data1$Question=="Recommendation to Career Field 36")],main="Q36 Distribution",xlab="polarity",ylab="Freq")
```

#Syuzhet Package

```{r}
library(syuzhet)
#get_sentiment(x vector of strings, )
syuzhet.v<-get_sentiment(data1$Response,method="syuzhet")
syuzhet.bing.v<-get_sentiment(data1$Response,method="bing")
syuzhet.afinn.v<-get_sentiment(data1$Response,method="afinn")
syuzhet.nrc.v<-get_sentiment(data1$Response,method="nrc")
syuzhet.m<-cbind(syuzhet.v,syuzhet.bing.v,syuzhet.afinn.v,syuzhet.nrc.v)
#creates place to plot 4 graphs as 2x2
#must be run as one chunk
par(mfrow=c(2,2))
hist(syuzhet.v)
hist(syuzhet.bing.v)
hist(syuzhet.afinn.v)
hist(syuzhet.nrc.v)
#scaling for comparison
syu.scale<-sign(syuzhet.m)
summary(syu.scale)
#recalculating with multi removed
```

```
syuzhet.v2<-
get_sentiment(data3$Response,met
hod="syuzhet")
syuzhet.bing.v2<-
get_sentiment(data3$Response,met
hod="bing")
syuzhet.afinn.v2<-
get_sentiment(data3$Response,met
hod="afinn")
syuzhet.nrc.v2<-
get_sentiment(data3$Response,met
hod="nrc")
syuzhet.m2<-
cbind(syuzhet.v2,syuzhet.bing.v2
,syuzhet.afinn.v2,syuzhet.nrc.v2
)
#scaling for comparison
syu.scale2<-sign(syuzhet.m2)
```
#meanr
```{r}
library(meanr)
meanr.v<-score(data1$Response)
meanr.v<-meanr.v$score
summary(meanr.v)
hist(meanr.v)
meanr.scale<-sign(meanr.v)
#re-computing
meanr.v2<-score(data3$Response)
meanr.v2<-meanr.v2$score
meanr.scale2<-sign(meanr.v2)
```
#sentimentr
```{r,warnings=FALSE,message=FAL
SE}
library(sentimentr)
sentr.v<-
sentiment_by(data1$Response)
sentr.v<-sentr.v$ave_sentiment
#sentr.v2<-
sentiment_by(data1$Response,aver
aging.function=average_mean)
#sentr.v2<-
sentr.v2$ave_sentiment
#summary(sentr.v)
#summary(sentr.v2)
par(mfrow=c(1,2))
hist(sentr.v)
hist(sentr.v2)
qplot(data1$sentiment_by,
geom="histogram", binwidth=0.1,
main="Review Sentiment
Histogram")
ggplot(data1,aes(x=sentr.v))+geo
m_histogram()+facet_grid(~Questi
on)+theme_bw()
```

```
sentr.scale<-sign(sentr.v)
sentr.v2<-
sentiment_by(data3$Response)
sentr.v2<-sentr.v2$ave_sentiment
sentr.scale2<-sign(sentr.v2)
```

#SentimentAnalysis Package

```{r}
library(SentimentAnalysis)
library(tm)
#initial analysis with default
parameters
scale<-sign(AS.m1a)
AS.scaleb<-sign(AS.m1b2
#Improvement Attempt 1:
stopwords with pronouns
#manually pre-process since
automatic removes stop words
eng.stopwords<-
stopwords("english")
#create list of words to *not*
remove
pronouns.we<-
c("we","our","ours","ourselves",
"we're","we've","we'd","we'll")
pronouns.i<-
c("i","i'd","i'll","i'm","i've",
"me","my","myself")
AS.m1<-
analyzeSentiment(data1$Response)
AS.m1a<-AS.m1[,c(2,5,8,12)]
AS.m1b<-
analyzeSentiment(data3$Response)
AS.m1b2<-AS.m1b[,c(2,5,8,12)]
#want columns SentGI, SentHE,
SentLM, SentQDAP
#11 is LM Uncertainty
#2,5,8,11,12
AS.

pronouns.you<-
c("you","you'd","you'll","you're
","you've","your","yours","yours
elf","yourselves")
pronouns.they<-
c("they","they'd","they'll","the
y're","they've")
all.pronouns<-c(pronouns.we,
pronouns.i,pronouns.you,pronouns
.they)
#create new list of stopwords
new.stopwords<-
eng.stopwords[eng.stopwords %ni%
all.pronouns]
#reformat to feed to function
```

```
corpus <-
VCorpus(VectorSource(data1$Respo
nse))
tdm<-tm_map(corpus, removeWords,
c(new.stopwords))
dtm<-DocumentTermMatrix(tdm)
AS.m2<-analyzeSentiment(dtm)
AS.m2a<-AS.m2[,c(2,5,8,12)]
#add pronouns to dictionary
#adding to LM since that had the
highest correlation
data(DictionaryLM)
str(DictionaryLM)
dict.LM<-loadDictionaryLM()
dict.Pronoun.pos<-
c(dict.LM$positiveWords)
dict.Pronoun.neg<-
c(dict.LM$negativeWords,pronouns
.you,pronouns.they)
pos.l<-length(dict.Pronoun.pos)
neg.l<-length(dict.Pronoun.neg)
all.words<-
c(dict.Pronoun.pos,dict.Pronoun.
neg)
all.scores<-c(rep(1,pos.l),rep(-
1,neg.l))
dictionaryPronoun <-
SentimentDictionaryWeighted(all.
words, all.scores)
sent.pn <-
analyzeSentiment(dtm,rules=list(
"PronounSentiment"=list(ruleLine
arModel, dictionaryPronoun)))
sent.pn.scale<-sign(sent.pn)
T<-
table(sent.pn.scale$PronounSenti
ment,data1$MSNum)
acc<-sum(diag(T))/sum(T)
pacc.v<-T[1,1]/sum(T[1,])
nacc.v<-T[3,3]/sum(T[3,])
nnacc.v<-
sum(T[1,1],T[3,3])/sum(T[1,],T[3
,])
acc
pacc.v
nacc.v
nnacc.v

#Improvement Attempt 2: Air
Force Dictionary
#in order to identify words
maybe being taken out of
context, ran word contribution
to score
```

```
#word contribution to score
#colnames(matGen)
#colnames(matSA)
#comp.scores<-matGen[,7]
#matSA$SentimentGI
#omp.scores<-matSA$SentimentLM
comp.scores<-
sent.pn$PronounSentiment
dict<-
generateDictionary(data1$Respons
e,comp.scores)
dict
#adjust dictionary by seeing
which words are in dictionary
data(DictionaryLM)
str(DictionaryLM)
dict.LM<-loadDictionaryLM()
#returns character vector 0 if
word not in dictionary, else
returns top word
dict.LM$positiveWords[which("pos
it" %in% dict.LM$positiveWords)]
#remove posit and opportun from
LM dictionary, since they are
articles of interest and the
purpose of the survey is to
determine the polarity
positiveWords<-
dict.LM$positiveWords[-
which(dict.LM$positiveWords=="po
sit")]
positiveWords<-positiveWords[-
which(positiveWords=="opportun")
]
negativeWords<-
c(dict.LM$negativeWords,pronouns
.you,pronouns.they)
pos.l<-length(positiveWords)
neg.l<-length(negativeWords)
all.words<-
c(positiveWords,negativeWords)
all.scores<-c(rep(1,pos.l),rep(-
1,neg.l))
dictionaryWords<-
SentimentDictionaryWeighted(all.
words, all.scores)
words.sent <-
analyzeSentiment(dtm,rules=list(
"UniqueWords"=list(ruleLinearMod
el, dictionaryWords)))
sent.words.scale<-
sign(words.sent)
T<-
table(sent.words.scale$UniqueWor
ds,data1$MSNum)
```

```
acc<-sum(diag(T))/sum(T)
pacc.v<-T[1,1]/sum(T[1,])
nacc.v<-T[3,3]/sum(T[3,])
nnacc.v<-
sum(T[1,1],T[3,3])/sum(T[1,],T[3
,])
acc
pacc.v
nacc.v
nnacc.v
#details about given dictionary
#data(DictionaryGI)
#str(DictionaryGI)
#dict.GI<-loadDictionaryGI()
#summary(dict.GI)
#SentimentDictionaryWordlist
#topic1<-
SentimentDictionaryWordlist(c("l
eadership","mentor","boss"))
#summary(topic1)
#performance evaluation
compareDictionaries(dict,loadDic
tionaryQDAP())
par(mfrow=c(2,2))
plotSentimentResponse(syuzhet.m[
,1],data1$MSNum,ylab="ManualScor
es",xlab="syuzhet")
plotSentimentResponse(syuzhet.m[
,2],data1$MSNum,ylab="ManualScor
es",xlab="syuzhet.bing")
plotSentimentResponse(syuzhet.m[
,3],data1$MSNum,ylab="ManualScor
es",xlab="syuzhet.afinn")
plotSentimentResponse(syuzhet.m[
,4],data1$MSNum,ylab="ManualScor
es",xlab="syuzhet.nrc")
plotSentimentResponse(matSA[,5],
data1$MSNum,ylab="ManualScores",
xlab="SentQDAP")
compareToResponse(sent.pn,data1$
MSNum)
#performance optimization
sentiment<-
analyzeSentiment(data1$Response,
rules=list("SentimentLM"=list(ru
leSentiment,loadDictionaryLM()))
)
sentiment
```
#All methods compare
```{r}
#library("PerformanceAnalytics")
library("corrplot")
data3$AvgSent<-
as.double(data3$AvgSent)
```

```
matGen<-
cbind(data1$MSNum,syuzhet.m,mean
r.v,sentr.v)
colnames(matGen)<-
c("Manual","syuzhet","syu.bing",
"syu.afinn","syu.nrc","meanr","s
entr")
cor<-cor(matGen)
#chart.Correlation(matGen,
histogram=TRUE, pch=19)
matGen2<-
cbind(data3$AvgSent,syuzhet.m2,m
eanr.v2,sentr.v2)
colnames(matGen2)<-
c("Manual","syuzhet","syu.bing",
"syu.afinn","syu.nrc","meanr","s
entr")
cor(matGen2)
matGen.scale<-cbind(data1$MSNum,
syu.scale,meanr.scale,sentr.scal
e)
colnames(matGen.scale)<-
c("Manual","syuzhet","syu.bing",
"syu.afinn","syu.nrc","meanr","s
entr")
cor2<-cor(matGen.scale)
par(mfrow=c(1,2))
corrplot(cor,method="circle")
corrplot(cor2,method="circle")

matGen.scale2<-
cbind(data3$AvgSent,
syu.scale2,meanr.scale2,sentr.sc
ale2)
colnames(matGen.scale2)<-
c("Manual","syuzhet","syu.bing",
"syu.afinn","syu.nrc","meanr","s
entr")
cor(matGen2)
matSA<-cbind(data1$MSNum,
AS.m1a)
cor3<-cor(matSA)
matSA.scale<-cbind(data1$MSNum,
AS.scale)
cor4<-cor(matSA.scale)
matSA.scaleb<-
cbind(data3$Response,AS.scaleb)
compM<-matSA.scaleb
l<-dim(compM)[2]
acc.v<-rep(0,l) #total accuracy
pacc.v<-rep(0,l) #correct
positives
nacc.v<-rep(0,l) #correct
negatives
nnacc.v<-rep(0,l) #no neutral
accuracy
```

```
for (i in 2:l){
  T<-table(compM[,1],compM[,i])
  acc.v[i]<-sum(diag(T))/sum(T)
  pacc.v[i]<-T[1,1]/sum(T[1,])
  nacc.v[i]<-T[3,3]/sum(T[3,])
  nnacc.v[i]<-
sum(T[1,1],T[3,3])/sum(T[1,],T[3
,])
}
acc.v
pacc.v
nacc.v
nnacc.v
colnames(compM)

#column 13, 14 only has neutrals
and positives classified
```
#Logistic Regression on each
word
```{r}
#For example, we can use the
broom package to perform
logistic #regression on each
word.
#library(broom)
#models <- inaug_freq %>%
#  group_by(word) %>%
#  filter(sum(n) > 50) %>%
#  do(tidy(glm(cbind(n,
year_total - n) ~ Year, .,
#              family =
"binomial"))) %>%
#  ungroup() %>%
#  filter(term == "Year")
#models
```,
#Topic Modeling
```{r}
#quest<-q27
q27<-
data1[which(data1$Question=="Car
eer Planning/Succession Planning
27"),c(1,3)]
q34<-
data1[which(data1$Question=="Edu
cation & Training 34"),c(1,3)]
q35<-
data1[which(data1$Question=="Awa
rds and Recognition 35"),c(1,3)]
q36<-
data1[which(data1$Question=="Rec
ommendation to Career Field
36"),c(1,3)]
quest<-data[,c(1,3)]
```

```
colnames(quest)<-
c("doc_id","text")
english_stopwords <-
readLines("https://slcladal.github.io/resources/stopwords_en.txt", encoding = "UTF-8")
corpus <-
Corpus(DataframeSource(quest))
processedCorpus <-
tm_map(corpus,
content_transformer(tolower))
processedCorpus <-
tm_map(processedCorpus,
removeWords, english_stopwords)
processedCorpus <-
tm_map(processedCorpus,
removePunctuation,
preserve_intra_word_dashes =
TRUE)
processedCorpus <-
tm_map(processedCorpus,
removeNumbers)
processedCorpus <-
tm_map(processedCorpus,
stemDocument, language = "en")
processedCorpus <-
tm_map(processedCorpus,
stripWhitespace)
# compute document term matrix
with terms >= minimumFrequency
minimumFrequency <- 5
DTM <-
DocumentTermMatrix(processedCorpus, control = list(bounds =
list(global =
c(minimumFrequency, Inf))))
# have a look at the number of
documents and terms in the
matrix
dim(DTM)
# due to vocabulary pruning, we
have empty rows in our DTM
# LDA does not like this. So we
remove those docs from the
# DTM and the metadata
sel_idx <- slam::row_sums(DTM) >
0
DTM <- DTM[sel_idx, ]
textdata <- textdata[sel_idx, ]
# number of topics
K <- 4
# set random number generator
seed
set.seed(9161)
```

```
# compute the LDA model,
inference via 1000 iterations of
Gibbs sampling
topicModel <- LDA(DTM, K,
method="Gibbs",
control=list(iter = 500, verbose
= 25))
# have a look a some of the
results (posterior
distributions)
tmResult <-
posterior(topicModel)
# format of the resulting object
attributes(tmResult)
nTerms(DTM)
# topics are probability
distribtions over the entire
vocabulary
beta <- tmResult$terms   # get
beta from results
dim(beta)                # K
distributions over nTerms(DTM)
terms
rowSums(beta)            # rows
in beta sum to 1
nDocs(DTM)               # size
of collection
# for every document we have a
probaility distribution of its
contained topics
theta <- tmResult$topics
dim(theta)               #
nDocs(DTM) distributions over K
topics
rowSums(theta)[1:10]     # rows
in theta sum to 1
terms(topicModel, 10)
exampleTermData <-
terms(topicModel, 10)
exampleTermData[, 1:3]
# select to 40 most probable
terms from the topic by sorting
the term-topic-probability
vector in decreasing order
top40terms <-
sort(tmResult$terms[topicToViz,]
, decreasing=TRUE)[1:20]
words <- names(top40terms)
# extract the probabilites of
each of the 40 terms
probabilities <-
sort(tmResult$terms[topicToViz,]
, decreasing=TRUE)[1:40]
# visualize the terms as
wordcloud
```

```
mycolors <- brewer.pal(8,
"Dark2")
wordcloud(words, probabilities,
random.order = FALSE, color =
mycolors)
```

```{r}
library(LDAvis)
topicmodels_json_ldavis <-
function(fitted, corpus,
doc_term){
     ## Required packages
     library(topicmodels)
     library(dplyr)
     library(stringi)
     library(tm)
     library(LDAvis)
     ## Find required quantities
     phi <-
posterior(fitted)$terms %>%
as.matrix
     theta <-
posterior(fitted)$topics %>%
as.matrix
     vocab <- colnames(phi)
     doc_length <- vector()
     for (i in 1:length(corpus))
{
          temp <-
paste(corpus[[i]]$content,
collapse = ' ')
          doc_length <-
c(doc_length, stri_count(temp,
regex = '\\S+'))
     }
     temp_frequency <-
inspect(doc_term)
     freq_matrix <-
data.frame(ST =
colnames(temp_frequency)
Freq = colSums(temp_frequency))
     rm(temp_frequency)
     ## Convert to json
     json_lda <-
LDAvis::createJSON(phi = phi,
theta = theta,
vocab = vocab,
doc.length = doc_length,

term.frequency =
freq_matrix$Freq)

     return(json_lda)
}
```

```
term_tfidf <-
tapply(DTM$v/slam::row_sums(DTM)
[DTM$i], DTM$j, mean) *

log2(tm::nDocs(DTM)/slam::col_su
ms(DTM > 0))
summary(term_tfidf)
tmreduced.dtm <- DTM[,term_tfidf
>= 0.4418]
summary(slam::col_sums(DTM))
tm.model <-
topicmodels::LDA(DTM, 3, method
= "Gibbs", control =
list(iter=2000, seed = 0622))
tm.corpus <-
Corpus(DataframeSource(quest))
tm.topics <-
topicmodels::topics(tm.model, 1)
## In this case I am returning
the top 30 terms.
tm.terms <-
as.data.frame(topicmodels::terms
(tm.model, 30), stringsAsFactors
= FALSE)
tm.terms
# Creates a dataframe to store
the Lesson Number and the most
likely topic
doctopics.df <-
as.data.frame(tm.topics)
doctopics.df <-
dplyr::transmute(doctopics.df,
LessonId =
rownames(doctopics.df), Topic =
tm.topics)
doctopics.df$LessonId <-
as.integer(doctopics.df$LessonId
)
topicTerms <-
tidyr::gather(tm.terms, Topic)
topicTerms <- cbind(topicTerms,
Rank = rep(1:30))
topTerms <-
dplyr::filter(topicTerms, Rank <
4)
topTerms <-
dplyr::mutate(topTerms, Topic =
stringr::word(Topic, 2))
topTerms$Topic <-
as.numeric(topTerms$Topic)
topicLabel <- data.frame()
for (i in 1:27){
     z <-
dplyr::filter(topTerms, Topic ==
i)
```

```
     l <-
as.data.frame(paste(z[1,2],
z[2,2], z[3,2], sep = " " ),
stringsAsFactors = FALSE)
     topicLabel <-
rbind(topicLabel, l)
}
colnames(topicLabel) <-
c("Label")
topicLabel
theta <-
as.data.frame(topicmodels::poste
rior(tm.model)$topics)
head(theta[1:5,])
x <-
as.data.frame(row.names(theta),
stringsAsFactors = FALSE)
colnames(x) <- c("LessonId")
x$LessonId <-
as.numeric(x$LessonId)
theta2 <- cbind(x, theta)

## Returns column means grouped
by catergory
theta.mean.by <- by(theta2[,
c(2:28)], theta2, colMeans)
theta.mean <- do.call("rbind",
theta.mean.by)
#I can now correlate the topics.
library(corrplot)
c <- cor(theta.mean)
corrplot(c, method = "circle")
tm.json <-
topicmodels_json_ldavis(tm.model
, tm.corpus, tmreduced.dtm)
serVis(tm.json)
```

Bibliography

A., V., & Sonawane, S. S. (2016). Sentiment Analysis of Twitter Data: A Survey of Techniques. *International Journal of Computer Applications*, *139*(11), 5–15. https://doi.org/10.5120/ijca2016908625

Almatarneh, S., & Gamallo, P. (2019). Comparing Supervised Machine Learning Strategies and Linguistic Features to Search for Very Negative Opinions. *Information*, *10*(1), 16. https://doi.org/10.3390/info10010016

Araujo, Diniz, & Bastos. (2016, March 31). *iFeel 2.0: A Multilingual Benchmarking System for Sentence-Level Sentiment Analysis*. 10th International AAAI Conference on Web and Social Media. https://www.aaai.org/ocs/index.php/ICWSM/ICWSM16/paper/view/13039

Borcan, M. (2020, June 8). *TF-IDF Explained And Python Sklearn Implementation*. Medium. https://towardsdatascience.com/tf-idf-explained-and-python-sklearn-implementation-b020c5e83275

Carnahan, D. (2017). *Confessions of a Data Scientist | Text Analysis with R.* Retrieved January 29, 2021, from https://confessionsofadatascientist.com/text-analysis-with-r-repro-report.html

Chen, E. (2011). *Introduction to Latent Dirichlet Allocation*. Retrieved January 18, 2021, from https://blog.echen.me/2011/08/22/introduction-to-latent-dirichlet-allocation/

Clark, M. (2018). *An Introduction to Text Processing and Analysis with R*. Retrieved December 22, 2020, University of Michigan. https://m-clark.github.io/text-analysis-with-R/

Costa, A., & Veloso, A. (2015). *Employee Analytics through Sentiment Analysis*.

https://doi.org/10.13140/RG.2.1.1623.3688

Council. (2020, December 16). *Council Post: 11 Ways (And Reasons) To Measure The ROI Of Your Company Culture*. Forbes. https://www.forbes.com/sites/forbeshumanresourcescouncil/2020/12/16/11-ways-and-reasons-to-measure-the-roi-of-your-company-culture/

Dua, S. (2020, May 18). *Text Classification using K Nearest Neighbors*. Medium. https://towardsdatascience.com/text-classification-using-k-nearest-neighbors-46fa8a77acc5

Feuerriegel, S., & Proellochs, N. (2019, March 26). *Sentiment Analysis Vignette* [Cran.r-project.org].

Fuchs, M. (2020, December 28). *Doing your first sentiment analysis in R with Sentimentr*. Medium. https://towardsdatascience.com/doing-your-first-sentiment-analysis-in-r-with-sentimentr-167855445132

Guiso, L., Sapienza, P., & Zingales, L. (2012). The value of corporate culture. *Journal of Financial Economics*, 60–76.

Hatzivassiloglou, V., & McKeown, K. R. (1997). Predicting the Semantic Orientation of Adjectives. *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, 174–181. https://doi.org/10.3115/976909.979640

Huselid, M. A. (1995). The Impact of Human Resource Management Practices on Turnover, Productivity, and Corporate Financial Performance. *Academy of Management Journal*, *38*(3), 635–872.

Jagtap, V.S., and K. Pawar. “Analysis of Different Approaches to Sentence-Level Sentiment Classification.” *International Journal of Scientific Engineering and Technology*, vol. 2, no. 3, 2013, pp. 164–170.

Jipa, G. (2019). The Value of Structured and Unstructured Content Analytics of Employees’ Opinion Mining. *Journal of Administrative Sciences and Technology, 2019*, 1–25. https://doi.org/10.5171/2019.908286

Jotheeswaran, J. (2012, April). *(PDF) Feature Reduction using Principal Component Analysis for Opinion Mining*. https://www.researchgate.net/publication/250917402_Feature_Reduction_using_Principal_Component_Analysis_for_Opinion_Mining

Judd, S. (2018, March 14). Employee Surveys Are Still One of the Best Ways to Measure Engagement. *Harvard Business Review*. https://hbr.org/2018/03/employee-surveys-are-still-one-of-the-best-ways-to-measure-engagement

Kenny, G. (2020, September 14). *What Are Your KPIs Really Measuring?* Harvard Business Review. https://hbr.org/2020/09/what-are-your-kpis-really-measuring

Kiprono, K. W., & Abade, E. O. (2016). Comparative Twitter Sentiment Analysis Based on Linear and Probabilistic Models. *International Journal on Data Science and Technology*, *2*(4), 41. https://doi.org/10.11648/j.ijdst.20160204.11

Koch, K. (2020, March 26). *A Friendly Introduction to Text Clustering*. Medium. https://towardsdatascience.com/a-friendly-introduction-to-text-clustering-fa996bcefd04

Lu, Bin, et al. “2011 IEEE 11th International Conference on Data Mining Workshops (ICDMW) Pp 81-88.” *10.1109/ICDMW.2011.125*, 11 Dec. 2011.

Luo, N., Zhou, Y., & Shon, J. (2016). *Employee Satisfaction and Corporate Performance: Mining Employee Reviews on Glassdoor.com*. 16. 37th International Conference on Information Systems, Dublin, Ireland

Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, *5*(4), 1093–1113. https://doi.org/10.1016/j.asej.2014.04.011

Misuraca, M., Forciniti, A., Scepi, G., & Spano, M. (2020). *Sentiment Analysis for Education with R: packages, methods and practical applications*. 27. arXir: 2005.12840

Na, J.-C., Sui, H., Khoo, C., Chan, S., & Zhou, Y. (2004). *Effectiveness of Simple Linguistic Processing in Automatic Sentiment Classification of Product Reviews*. 8. Ed. I.C. McIlwaine. Knowledge Organization and the Global Information Society: Proceedings of the Eighth International 15KO Conference, Wurzburg, Germany, pp 49-54

Naldi, M. (2019). A review of sentiment computation methods with R packages. *ArXiv:1901.08319 [Cs]*. http://arxiv.org/abs/1901.08319

Pang, B., & Lee, L. (2008). *Opinion mining and sentiment analysis*. Foundations and Trends in Information Retrieval, Vol 2, No 2, pp 1-135

Pröllochs, N., Feuerriegel, S., & Neumann, D. (2018). Statistical inferences for polarity identification in natural language. *PLOS ONE*, *13*(12), e0209323.

https://doi.org/10.1371/journal.pone.0209323

Qiu, G., Liu, B., Bu, J., & Chen, C. (2009). Expanding Domain Sentiment Lexicon through Double Propagation. In *IJCAI International Joint Conference on Artificial Intelligence* (p. 1204).

Raja, A. (2017). Handling 'Happy' vs 'Not Happy': Better sentiment analysis with sentimentr in R. *DataScience+*. Retrieved January 29, 2021, from https://datascienceplus.com/handling-happy-vs-not-happy-better-sentiment-analysis-with-sentimentr-in-r/

Ribeiro, F. N., Araújo, M., Gonçalves, P., Gonçalves, M. A., & Benevenuto, F. (2016). SentiBench—A benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ Data Science, 5*(1), 1–29. https://doi.org/10.1140/epjds/s13688-016-0085-1

Salas-Zárate, M. del P., Medina-Moreira, J., Lagos-Ortiz, K., Luna-Aveiga, H., Rodríguez-García, M. Á., & Valencia-García, R. (2017, February 19). *Sentiment Analysis on Tweets about Diabetes: An Aspect-Level Approach* [Research Article]. Computational and Mathematical Methods in Medicine; Hindawi. https://doi.org/10.1155/2017/5140631

Salton, G., & Buckley, C. (2019). *Stopword List 2*. Lextek.Com. Retrieved January 18, 2021, from https://www.lextek.com/manuals/onix/stopwords2.html

Samuel, J., Ali, N., Rahman, M., Esawi, E., & Samuel, Y. (2020). COVID-19 Public Sentiment Insights and Machine Learning for Tweets Classification. *Information 2020 MDPI, 11*(314). https://doi.org/10.3390

Schweinberger, M. (2020, December 16). *Topic Modeling with R*.

https://slcladal.github.io/topicmodels.html

SMART stopword list. (1960). Retrieved January 18, 2021, from http://www.ai.mit.edu/projects/jmlr/papers/volume5/lewis04a/a11-smart-stop-list/english.stop

Snowball Stopword List. (1979). Snowball.Tartarus.Org. Retrieved January 18, 2021, from http://snowball.tartarus.org/algorithms/english/stop.txt

Staff, MonkeyLearn. (2020). *Everything There Is to Know about Sentiment Analysis*. Retrieved February 6, 2021, from https://monkeylearn.com/sentiment-analysis/

Stopword List 1. (2010). Lextek.Com. Retrieved January 18, 2021, from http://www.lextek.com/manuals/onix/stopwords1.html

Sund, A. E. (2017). *Employee Pronoun Use In Verbatim Comments As A Predictor Of Job Attitudes And Turnover Intentions*. 71.

Tan, S., & Zhang, J. (2008). An empirical study of sentiment analysis for chinese documents. *Expert Systems with Applications*, *34*(4), 2622–2629. https://doi.org/10.1016/j.eswa.2007.05.028

Varghese, R., & M, J. (2013). *A Survey on Sentiment Analysis and Opinion Mining*. International Journal of Research in Engineering and Technology, Vol 2 Issue 11, Nov 2013

Wang, N. (2017, November 21). *Topic modeling and sentiment analysis to pinpoint the perfect doctor*. Medium. https://blog.insightdatascience.com/topic-modeling-and-sentiment-analysis-to-pinpoint-the-perfect-doctor-6a8fdd4a3904

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
03/25/2021	Master's Thesis	September 2019 - March 2021

4. TITLE AND SUBTITLE
Automated Sentiment Analysis for Personnel Survey Data in the US Air Force Context

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Haines, Julia M, Ms, GS-09

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
Wright-Patterson AFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT-ENS-MS-21-M-164

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Intentionally Left Blank

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution Statement A. Approved for Public Release;Distribution Unlimited

13. SUPPLEMENTARY NOTES
This work is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT
When surveys are distributed across the Air Force (AF), significant resources are put towards their development, distribution and analysis. However, when open-ended questions are included, respondent comments are generally under-utilized. This is due to a lack of transparency and confidence in the accuracy of machine-aided methods such as sentiment analysis and topic modeling. No model or methodology has been universally identified as ideal for this use case. This research quantifies the accuracy of some common sentiment analysis methods in order to gain a better understanding of the scope to which they can be applied.

15. SUBJECT TERMS
Sentiment Analysis, Opinion Mining, Topic Modeling, Survey, Text Analysis, Lexicon

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Lance Champagne, AFIT/ENS
					19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UU	100	937-255-6565 lance.champagne@afit.edu