2020

Passive Data Collection Practices

Garfinkel, Lillian
Scheufele, Joseph
Staw, Benjamin
Whichard, Wayde
Worcester Polytechnic Institute

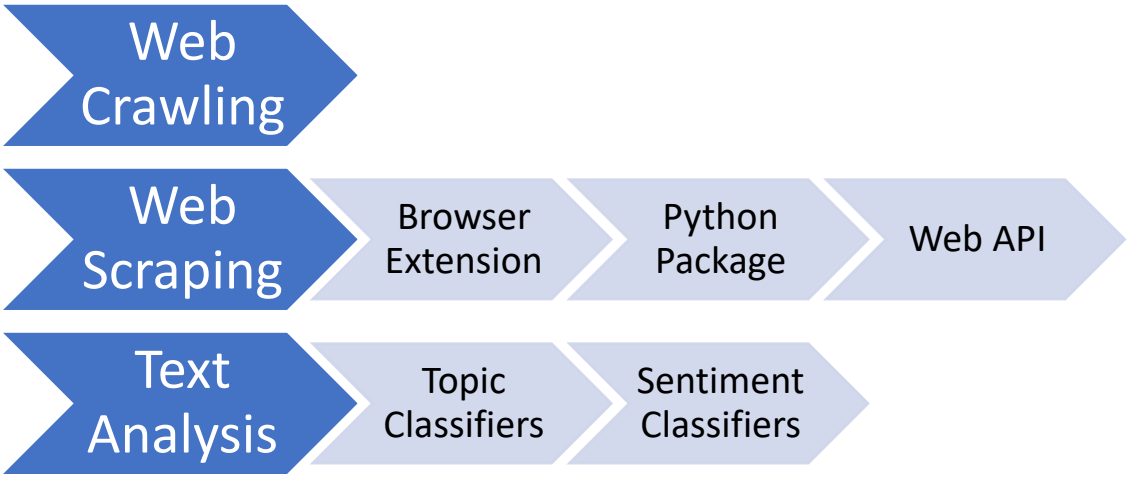
12/11/2020

TABLE OF CONTENTS

LITERATURE REVIEW OF PASSIVE DATA COLLECTION PRACTICES

Passive data collection (PDC) practices collect data without explicit interaction from the source. The USPTO was considering employing PDC to aid in the office's understanding of its customers' perceptions about the application process. Specifically, these perceptions depend on customer sentiment, satisfaction, and QoE (quality of experience). This literature review discusses several PDC strategies, analysis strategies, and associated open-source software. The conversation will include a review of each data collection method's effectiveness, how their results can be analyzed, their relevance to customer perceptions, and possible concerns of their use. This review of practices contributes to a recommendation for the USPTO about employing PDC strategies. For assistance in understanding the PDC process please look to Figure 1 below.

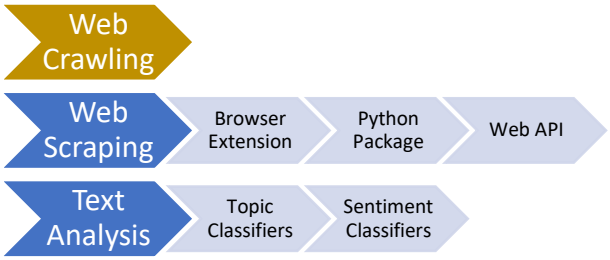
Figure 1: Literature Review Flowchart



As the literature review progresses certain sections will be highlighted to illuminate where in the process each technique belongs.

DATA COLLECTION STRATEGIES

Organizations can collect data to assist in understanding their customers. Instead of only conducting surveys and questionnaires, organizations can collect data online. PDC is a method of collecting data without direct involvement from a customer. Therefore, collecting passive data offers an opportunity to broaden an organization's customer sentiment database. The following sections will discuss practices of PDC. These practices include data mining, text mining, web scraping, web crawling, and web APIs. The discussions will include how the technique is conducted, how different organizations have used each technique, and the ethics of each practice. Following this discussion, there will be a brief analysis of how each method, software, or collection style could lead to a possible design.



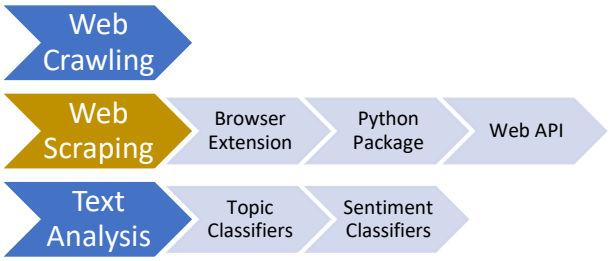
WEB CRAWLING

Sometimes organizations require many different perspectives from many locations to ascertain customer sentiment, satisfaction, and QoE. This can be achieved using a web crawler. Starting at an initial specified URL, the web crawler will traverse the whole webpage for other related URLs. It indexes the information it finds at each link, then proceeds to other related pages to do the same. The crawler itself does not analyze any information, it just constructs the path of pages for another method to collect and then analyze[1].

Web crawlers are popular among many industries as they are not limited to collecting data from one specific location. One source looked at over 250,000 reviews over three months gathered from multiple hotel review sites. They deployed a web crawler that started off looking at TripAdvisor for reviews, then branched off to other hotel websites from there[2]. In another case, a web crawling tool was

deployed on a transit customer blog page. This tool was able to gather highlights, setbacks and recommendations and quantify the riders' experience with the train service[3]. In another instance, Lyu and Choi developed a Python-based web crawler as part of research to predict sales volume in the organic food industry. This crawler simulated a browser's access to the internet and automatically collected the content of web pages related to organic products. It then broke the web pages down to only the content of interest, specifically their reviews. The collection period spanned 10 days and it collected half a million reviews from 9040 products[4].

These examples have one prominent commonality. The web crawlers were implemented to measure some form of customer perceptions. In a more specific case, Chang used a web crawler to collect synonyms to words like 'customer', 'vendor', and 'trading' from dictionaries or related documents to build a lexicon. They then used this lexicon to hunt for words within customer reviews that are also in the lexicon. This method is time consuming, as the crawler needs to search through the lexicon for every word[5].



WEB SCRAPING

Web scraping is a method of extracting data from the internet. Typically, the collected data is returned as a CSV or similar format so that it can be stored in a convenient way[6]. Although web scraping can be carried out manually, the process is usually automated. Web scraping using automation tools increases the efficiency at which data can be collected from the internet[7]. One major issue to avoid with web scraping is committing a DNS (Denial of Service) attack. This means that the user made too many requests to a web server in too short a time and can result in the user being sued.

Websites store data in a format called hypertext mark-up language (HTML). HTML uses tags and rules to organize and provide information about a website's data. HTML is also written as plain text, so it appears in a text editor in the same form that it is interpreted by software[8]. The syntax rules of HTML follow node, attribute, and text rules:

1. Each element must have opening and closing tags <element> </element> and can be nested.
2. Text is contained between the element nodes.
3. Attributes of each element are inside the opening element node and must be quoted.

Figure 2: Sample HTML code with the website it generates.

```
<!DOCTYPE html>
<html>
  <head>
      <meta charset='utf-8'>
      <title>Test</title>
  </head>
  <body>
      <h1>Header</h1>
      <h2>List of Items</h2>
      <ul>
        <li class='something'>
          Text
        </li>
        <li>
          Some More Text
        </li>
      </ul>
  </body>
</html>
```

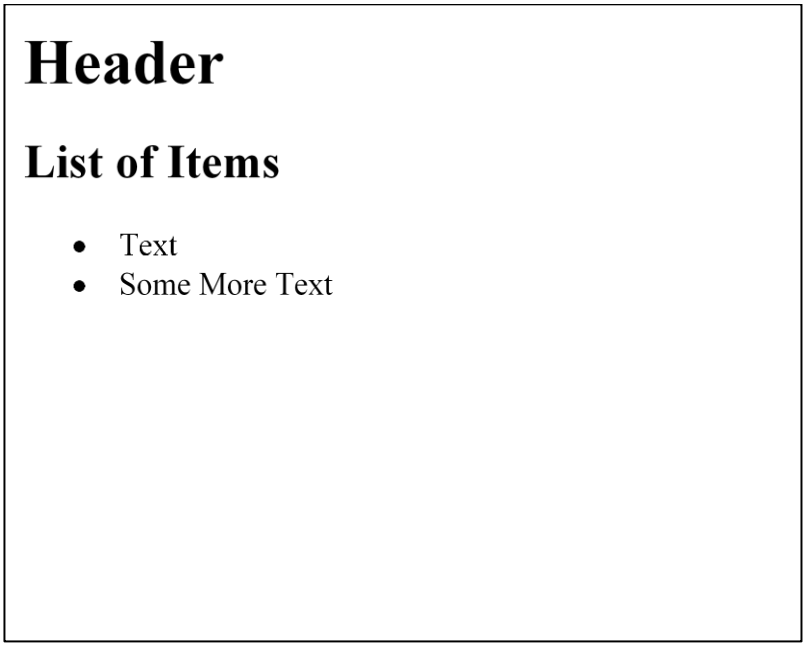


Figure 2 is a sample of HTML code to demonstrate how a website is constructed and how what is on the left side is interpreted on the right side.

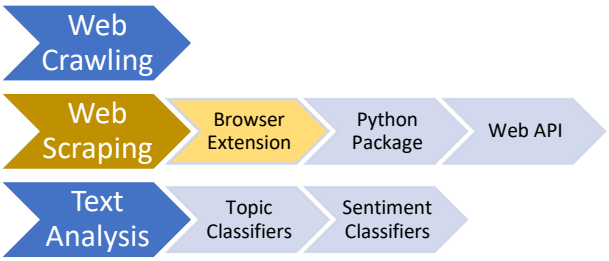
Figure 2 is an extremely basic sample of a website, however the principles of web scraping still hold. If a computer program is to extract information from this website hosted on the internet, it must request that information from the website's web server. This request is communicated through XPATH[8]. XPATH is the language meant for querying data from a web site. XPATH operates by leveraging HTML's tree-like structure. The left image of Figure 2 above can be found in a web browser if "inspect element"

is turned on. Within the inspect element window of a browser there exists a console that can take an input of XPATH code and return an output of the specified information. An example of XPATH would be:

```
>>  $x(//li[@class = 'something'])
```

The above code will look anywhere in the HTML of the current web page for the '<li>' element with class equal to 'something.' This query returns an array of all the elements that match this criterion. Each array node carries all the information held within the HTML element. More specific information on how to extract only the text from the elements and other commands are available in open-source documentation[8]. Manually entering this code into the console is tedious. Luckily there exists plenty of open source and paid software that can automate the process in different ways.

WEB SCRAPING SOFTWARE



Different forms of web scraping software require differing levels of human interaction, programming expertise and ease of use. The first kind of web scraping tools are a family of web extensions. A web extension is an application that mounts onto a user's web browser that adds functionality[9]. In this case, the user gains the ability to click their mouse over content on a website. The web extension will then locate where in the HTML code the user's click represents and extract the data from the selected element. One example of a web scraping web extension, is AnyPicker. AnyPicker is available on the Google Chrome web store. When AnyPicker is turned on, the user will click and drag a box over the areas from which they want to collect data, as seen in Figure 3.

Figure 3: AnyPicker Screenshot

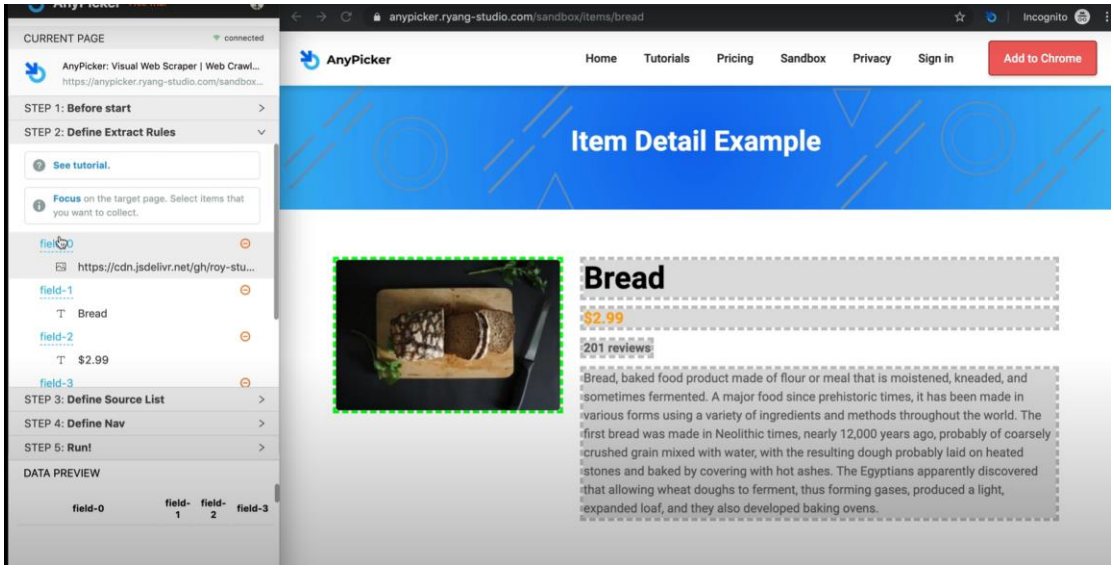
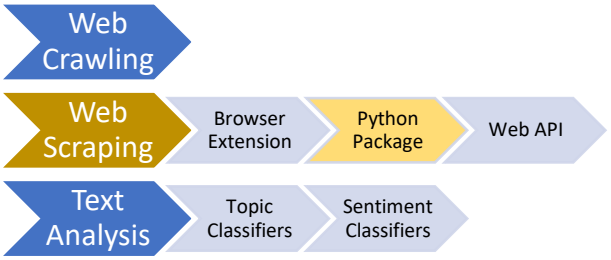


Figure 3 is a screenshot of the AnyPicker extension scraping text from a website.

The grey boxes on the right of Figure 3 represent HTML elements with their text data. On the left of the screen is the set up for the web scraper. The extraction rules essentially lay out what the scraper will collect. Next the data source list must be set. Most web extension web scrapers rely on the HTML being standardized (i.e. website structure is the same). Thus, the data sources are most likely a list of pages from the same website with the same structure. Finally, the web scraper runs and collects the specified information from all the data sources specified and places them into a csv file. An additional feature included with AnyPicker is a timer and request limiter to control how often the scraper accesses each web server.

When searching for 'scraper' in the Google Chrome web store, many products appear. All of them will operate in the same way as AnyPicker. The key areas to look for when evaluating a web extension web scraper are limits on the number of pages that can be scraped per month, free-trial periods, quality of user interface and ability to save the data to a database or spreadsheet. Web extensions are the easiest to implement; all that is required is a single download and the user can begin

scraping. Little to no programming experience is required to operate this tool. The user does not need to know how to use XPATH. The problem with these web extensions is that they do not allow the user to customize the application and therefore cannot be modified to suit different needs. Customizations to include machine learning or text mining are not allowed. Web extension web scrapers offer a very easy and quick way to web scrape, however they are not the most efficient. Another issue with web extensions is being able to reach the software developer when there is an issue with the web extension.



Another type of web scraping involves creating a native application using an API (application programming interface). The most popular way of creating this application is with Python and a web scraping Python package. One major advantage to building a native app is it allows for a streamlined system. It allows one program to web crawl and scrape without having a human intervene to run separate programs. A web scraping Python app would be one part of this program. Another advantage is it allows for upgrades or changes to the system. Web scraping Python packages can be found on pypi.org, a repository for open-source projects, to be shared in a format that can be easily installed and added to any Python program. The source code for these packages can be found on the developer's GitHub (a community for code sharing)[10] or in the documentation supplied to pypi.org[11]. The two most popular web scraping packages available include Scrapy and Beautiful Soup.

Scrapy is more well-rounded while Beautiful Soup is less maintained and does not have as many features as Scrapy. Scrapy can web crawl and web scrape. It also has its own HTML file parser and can extract specific data. Beautiful Soup relies on a slow and outdated Python default parser. Essentially, Beautiful Soup creates a Python object that extracts the HTML data and does not require the user to

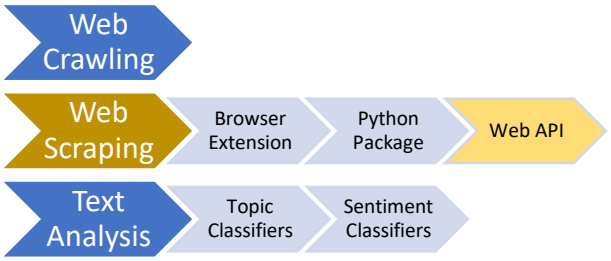
know XPATH[12]. This process can result in wasted resources and a slower program, because the object contains the entire website's data when most users are only looking for a few details. However, an advantage to this is if the user needs to make multiple requests to a website, this Python object can be accessed instead, thus reducing the load on the web server. Despite its other advantages, Scrapy does require XPATH knowledge. Using XPATH will reduce the size of the data being collected as the user controls exactly what parts of a website are to be collected. Scrapy has a native HTML extractor that is faster and more accurate than the default. Within Scrapy, the user creates a 'spider' object that is supplied a list of URLs, and the definition of a 'parse item.' The parse item contains the rules for what data are to be extracted[13].

Software like AnyPicker and Scrapy can have any URL as their input. Therefore, a user could enter the URL of a Google search, then scrape for any keywords related to customer perceptions within the results of that search. If this program finds a relevant element, then it can store the URL in a list for use by another web scraping program. The result is an automation of the web crawling process.

WEB SCRAPING APPLIED TO THE USPTO

Many organizations interact with their customers on blog pages or social media. Often, organizations will use web scraping to identify common trends in sentiment, satisfaction, and QoE. Web scraping finds meaning in patterns of human behavior[14] and can magnify hidden trends throughout the internet[15]. Web scraping is conducted in two parts: collecting data, then making a structured view of that data[16]. Three common approaches to web scraping are the weight measurement approach, differential approach, and machine learning approach[17]. The weight measurement approach picks a place in a web page and uses its code structure to find nearby words. The differential approach recognizes that most web pages share a large amount of non-content code such as formatting. Ignoring these similarities allows the tool to scrape more pages faster. The machine learning approach works by

introducing many examples of page structure to the tool until it can recognize the structure on its own. The tool will then be able to optimize how it scrapes pages with familiar structures[17]. A visual web scraper (such as AnyPicker, described above) can collect data anywhere a user clicks. Xu, Liu and Gursoy used a visual web scraper to collect customer reviews from airline booking websites. Visual web scraping makes it easier to scrape data from different sites that do not have similar code structures[18]. Web scraping consumes time and resources, especially when performed manually. As a result, it is important to be able to intelligently scrape web pages using the above methods. If an organization requires a large amount of predictable data to be gathered on the web, web scraping will be able to collect that data.



WEB API

Organizations may want to collect a large portion of data from a few select websites. When conditions are correct, this process can be expedited with the use of an application programming interface (API).

APIs span a wide range of services. An API could be a python package, it can also be a connection to a website that delivers information. For the purposes of this project, we will refer to Web APIs as a service provided by a website that delivers information upon request. For example, the twitter API allows users to query their database of every public tweet ever made[19].

Once code is written to direct an API on actionable information, it can help researchers find trends in data. For example, a user can set an API to notify them every time someone posts on Twitter about intellectual property. A web API will constantly query the server until it finds text related to

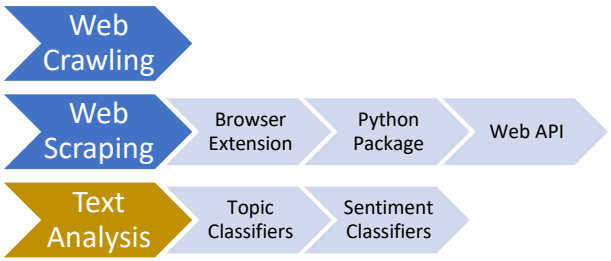
intellectual property. Once this request is captured by the API, the user's code can check the response from the API to see if a new post related to intellectual property has been made[20].

The USPTO can use an API when looking for posts made regarding the patent application process. Patently-O, IP-watchdog, Reddit, and Twitter all have APIs that allow them to quickly gather relevant comments. The API method of collecting data is very user friendly and may lead to deeper insight on customer expectations. APIs will work to collect this information directly from sources such as websites, databases, social media, and more. As a result, it is a very versatile data collection strategy. If the office were to adopt this method, there would not be many constraints on the data sources. The only limits of data collection via API is the accessibility of information. If there happen to be no comments, feedback or responses, the API will not have information to collect.

ETHICS OF DATA COLLECTION STRATEGIES

As seen above, data collection has the potential to make a large impact. If practiced unethically, it can also cause a large amount of harm. For example, large scale data collection is currently used to understand credit worthiness of consumers. Even if most consumers are not fully aware of what is happening, some argue that privacy is being infringed[21]. Privacy can also be reduced during counterterrorism efforts when many people's data is collected to learn more about a smaller subsection of criminals[22]. In some cases where there is a large amount of data to be collected and analyzed quickly, that data is collected autonomously. This causes a privacy concern as the autonomous collector makes the decision whether to collect specific data and does not always recognize when it is appropriate to proceed and when it is not[23]. People do not always realize how much data they are providing to big corporations. When many people realize how much data they are giving away, they become uncomfortable. This limit is often reached without some people realizing[24]. Given that the USPTO's goal is to use PDC to learn about its customers' perceptions with the intent of improving that perception, it is

important that attention is given to the amount of data being passively collected. Given that excessive data collection is perceived negatively, the USPTO can better protect its own image by being intentional about the data it collects.

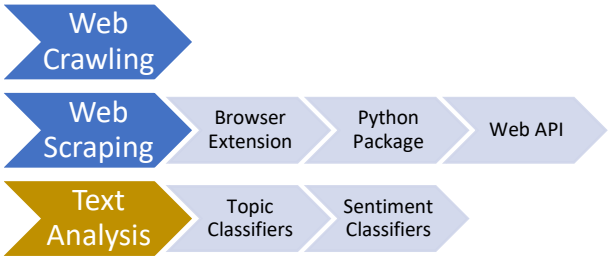


DATA MINING

Computer scientists have developed a variety of strategies and techniques to autonomously understand large sets of data. Christopher Clifton, Professor of Data Science at Purdue University, describes data mining as the process of discovering interesting and useful patterns and relationships in large volumes of data[25]. These patterns and relationships come directly from spreadsheets, online forums, social media and surveys. Data mining is used by companies to filter large amounts of raw data into categories that can be analyzed[26]. Large companies can use data mining to improve marketing of their products by mining user preferences[21]. These preferences can then be further analyzed to help an organization understand its customers.

In many cases, Data Mining is effective, yet can potentially harm consumers when personal information is bought and sold. This is known as consumer lurking[21]. An example of consumer lurking is data mining for counterterrorism efforts. Finding individuals who have characteristics or behaviors of known terrorists can be useful for preventing attacks but mining the population excessively can make

society less safe by exposing too much information about non-criminals[22]. Overall, data mining is safest when the information being collected for review is initially public or posted.

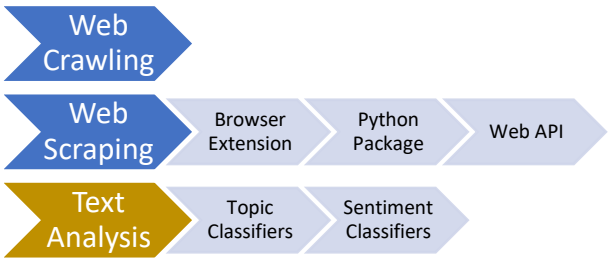


TEXT MINING

A more specific data mining technique for going through large sets of textual data is text mining. This is a method where large sets of text are filtered down to the phrases and words that are most prevalent within the text. These extracted words can then be further analyzed with more distinct techniques to create meaning within the data[27]. The two main strategies within text mining were categorizing phrases and parsing to remove unnecessary words.

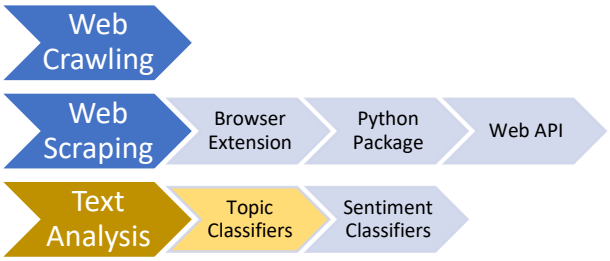
The team found that many sources reduce the size of the data set by removing words with no sentimental value. Liao and their team used an open-source software written in R to mine pre-transcribed text from customer insurance calls. They represented their preliminary data in a word cloud to visualize the words that appeared the most frequently. Because most of these words held no sentimental meaning, the text had to be preprocessed to remove these words before further analysis[28]. Liu and the team developed three basic steps for their text mining. First, a Python kit was used to separate the sentences into words. Next, they preprocessed the text for words with no sentimental meaning, and lastly, they merged the data for words that were synonyms[29]. When conducting text mining, Xu Xun dropped words that do not carry much information in the English language, such as “the”, “a”, “an”, “is”, etc. Xun also used root words to sort for and organize similar words. Words like completed, completely, and completing, all fall under the root of complete. The effect of these actions was a drastic reduction in computational time with very limited changes in results[30].

Text mining involves parsing text to collect keywords. Yoon et. al gathered their customer feedback sources from manufacturers' websites and surveys in XML documents. They then determined the specific keywords that referenced the topics they were looking for. Next, an open-source sentiment analysis software was deployed to evaluate the data[31]. Guo, Barnes and Jia cleansed their collected data and then parsed the text to remove non-English characters, low frequency words, and parts of speech to prepare the dataset for analysis[2].



DATA ANALYSIS STRATEGIES

Organizations must employ a data analysis technique that properly suits their data collection strategy, types of data and expected results. The types of data in question are customer reviews on social media, blog posts, and discussion topic websites. Analyzing this passive data provides insights into customers' expectations and perceived experience. Knowing customers' expectations and perceived experience will identify the organization's customer sentiment, satisfaction and quality of experience. The following is a discussion of data analysis techniques specifically related to their performance with textual customer review data. These techniques include Latent Semantic Analysis, Latent Dirichlet Analysis, Link Analysis, Aspect-Level Analysis, Naive Bayes Classifiers, and Support Vector Machines.



LATENT SEMANTIC ANALYSIS

Latent Semantic Analysis (LSA) is an analysis method used for finding links between words, and contextualizing words. It does this by defining the words of a document by their context rather than true definition[32]. It is capable of simulating human phenomena such as learning vocabulary words, word-categorizing, recognizing words derived from others, understanding conversations, and judgements of essay quality[33]. Landauer and his team studied LSA and tested how well it can distinguish words relative to a human. When prompted to select the best synonym to a given word, out of 4 choices, LSA was 65% accurate. This result came after training the system with over 4.5 million words from 30,000 encyclopedia articles[33]. Landauer saw the potential that LSA had in 1998, but his ability to improve upon this accuracy was limited due to the technology of the time. However, in 2020, technology has advanced to the point where more training data can be used with more efficient algorithms to increase the accuracy of LSA into the 75-85% range.

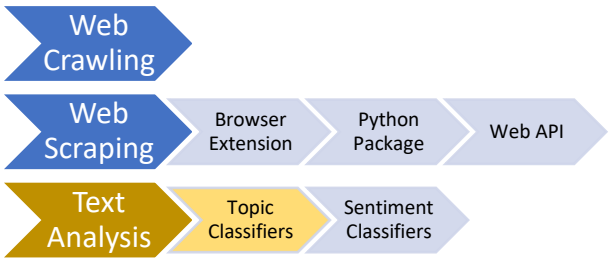
The first step in LSA is to represent the text as a matrix where each row stands for a unique word, and each column stands for a text passage. The cells of the matrix contain the frequency that the unique word appears in each passage. Once the matrix is created, singular value decomposition (SVD) is applied to the matrix, why this works is beyond the purview of this project. The product of conducting SVD is a least-squares best fit of the frequencies. Basically, LSA can predict how often words would appear in a text, even if they have yet to do so. That fact can be used to find the subtext of what is being written[33].

Latent Semantic Analysis models can be created with the fitlsa function in MATLAB. MATLAB is a trusted industry standard; it is used by million-dollar corporations and universities alike. All that is required for inputs is a matrix of frequencies, and the number of topics. The function generates a least squares best fit matrix[34]. LSA can also be conducted using the PLSA Python package. A function can be created that is like fitlsa, with the same inputs and outputs[35].

Through our research the team found 4 sources that utilize LSA for understanding customer perceptions. Xu Xun employed LSA to determine whether traveler satisfaction differed between different groups of people. He chose LSA for its ability to process an overload of data. It works in a manner like a human brain and can handle ambiguity very well[30]. In another study, Xun investigated if the factors that determine customer sentiment reflect how customers rated a hotel's services. He broke up the collected comments into groups based on the star rating of the comment. Then they used LSA to find if the star of the review matched the context of the comment[36]. Trappey and his team used LSA to infer purchase intentions of customers through scanning social media posts, blogs and other networking platforms. Trappey was targeting customers who specifically commented on a particular smartphone. The relationships between words in each comment are measured by the frequencies of the words, which ultimately eliminates any interpretive human error[32]. Mancolin and his team used LSA to understand customer perceptions of hotel services. They used 26,141 comments over 5 years. As with text mining, words with no meaning were removed before processing. Another factor they considered was Zipf's Law, which predicts that about 60% of the words appear only once in the data set and so were removed[37].

APPLICATIONS OF LATENT SEMANTIC ANALYSIS TO THE USPTO

Latent Semantic Analysis can provide the USPTO with topics that customers are talking about in their survey responses or beyond. These topics would be generated by considering the context of the surrounding words. Since some forms of LSA require training, and others do not[32, 33] , LSA may not require as much set up time, but can still produce similar results to other methods.



LATENT DIRICHLET ALLOCATION

Latent Dirichlet Allocation (LDA) is a probabilistic topic modeling technique. An LDA algorithm takes in a set of documents containing words. It then calculates the joint distribution of a topic mixture, a set of topics, and a set of words. Each document is modeled as a certain combination of topics, and each topic is modeled as a certain combination of words within the documents. Associating words with topics allows trends and commonalities between documents to become more apparent. This model provides a finer level of specificity than just considering the words alone in a document. The probabilities of topics among documents are then calculated using LDA to provide a representation of the contents of the documents[38]. These topics and the words they encompass can then be further analyzed for specific sentimental attributes.

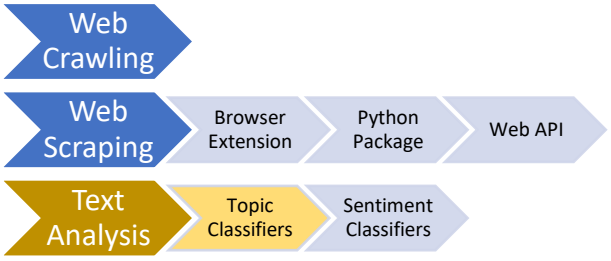
LDA has two significant uses: data analysis and data trend predictions. For example, when looking for specific information or data, there may be a lot of unnecessary or misleading information in the data set. To aid with minimizing misleading data, Liu and the team found satisfactory results when training a sentiment analysis algorithm to identify positive and negative connotations within text. They hand-selected 2000 of the 107,000 reviews to label as examples, then were able to successfully process

their dataset with the LDA based algorithm[29]. Additionally, Guo, Barnes and Jia utilize an LDA model to predict the most frequently occurring topics within the text they were analyzing. Each document is assumed to have multiple topics embedded within it. Then, the proportions of each topic within each document are evaluated to determine the relevance of said topic[2]. Lastly, Liao and their team used LDA to predict topics as well. They describe LDA as assuming each word in each document belongs to one of the underlying topics within the text meaning[28]. Lock and Pettit used LDA to extract the most relevant topics from their dataset of tweets. This process iterates through each word and associates similar words with the same topic[39].

In addition to finding trends, there were various methods of data analysis using LDA. An LDA model is capable of accurately predicting whether a new comment belongs to a certain topic and whether it is positive or negative, once the frequencies of the words are determined and the model is trained with the correct data[4]. Lyu and Choi believe LDA is exceptional at procuring topic and subject words from customer reviews. Bi and his team used LDA for extracting customer-sentiment-dimensions. These are the factors that affect a customer's sentiment. The dimensions are determined based on the probability distribution of words in the document. However, before using LDA, the textual data needed to be cleaned for meaningless words, which was time consuming. Separate programs were written purely for the preprocessing of the data[40].

APPLICATIONS OF LDA TO THE USPTO

One major element of the USPTO collecting customer data is being able to predict trends. LDA can discover these hidden trends in big data. If the USPTO were to use a method like LDA, trends in data would be easier to find. Additionally, trends could then be analyzed to fit the expectations of QoE regarding the application process. One issue to note is the training requirement, if a pretrained version cannot be found then the set up for LDA could be more time consuming.



SUPPORT VECTOR MACHINES

Support Vector Machines (SVMs) are a type of neural network that can be trained and programmed to classify textual inputs into categories. When applied to a large data set these classifications can establish the general topics or emotions of what is being said[41]. Grljevic and Bosnjak found that their SVM algorithm was about 80% effective in predicting negative comments and about 76% effective in predicting positive comments taken from a variety of review websites, like IMDb and Yelp. In all trials, the algorithm was more successful at identifying negative connotations than positive[42].

Support vector machines are designed to find the hyperplane of best fit. The first step is to translate the text data into frequency data. The hyperplane results in a boundary between the frequency data points. The size and shape of the boundary depends on the dimension. The boundary created by the hyper plane is called the decision boundary. Any input will be classified by its position relative to the boundary. For example, in 2-D a hyperplane is a line, if an input value falls above the line it is classified to one category, if it falls below the line, then it is classified to the other category[43, 44]. As shown in Figure 4, the 2-D hyperplanes do not need to be straight lines, they can be circular, quadratic

or exponential in shape. Support vector machines are largely controlled by their kernel function. The kernel function is a starting point for the system to converge to the optimal hyperplane shape.

Figure 4: 2-D best-fit hyperplanes

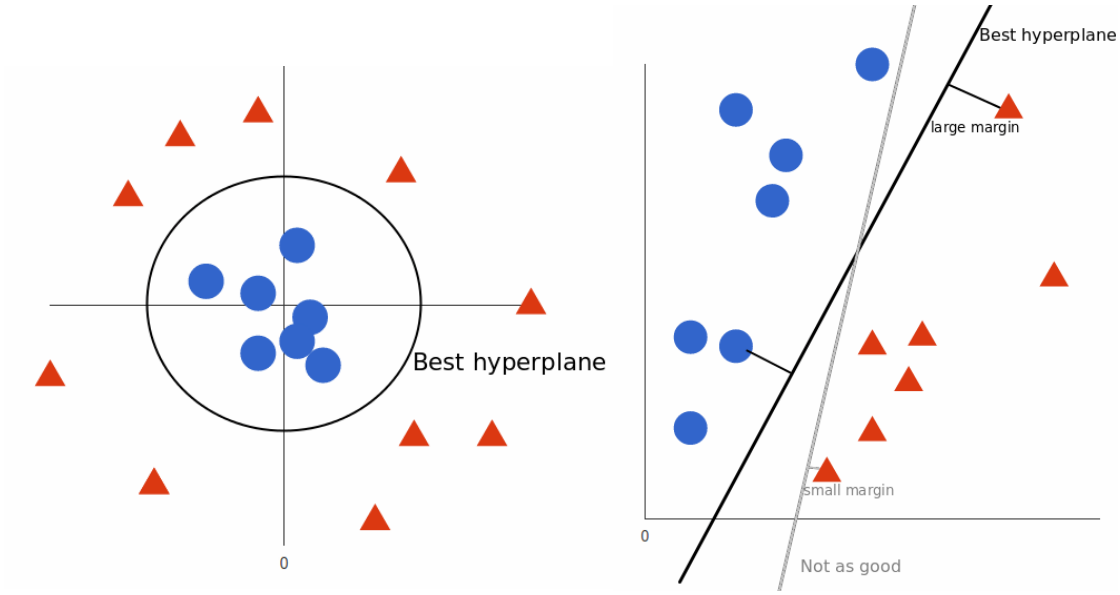


Figure 4 is two examples of a hyperplane converging to a particular shape of line. Left is converging to a circle; the right is a line.

One open-source library of support vector machine software is TensorFlow. A project created by Mozilla; it houses an API that can create an SVM. To work, the input dimension, options to tune the machine, kernel function, training data and prediction data are required. TensorFlow is a popular tool used by Twitter, GE, Coca Cola, and Google[45].

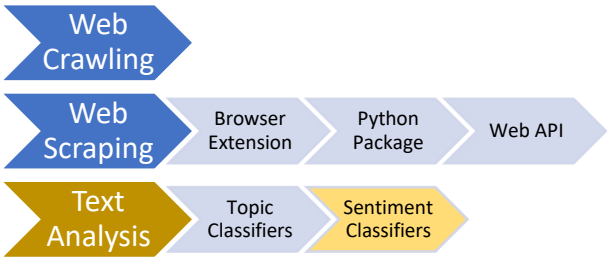
Another open-source library for SVMs is LIBSVM. If all that is required is to build an SVM with the most ease of use, then LIBSVM is a great choice. It does not have the unnecessary features of TensorFlow. Therefore, it will take up less space on a computer and run faster. LIBSVM uses a cache to store results from previous iterations of the SVM. Torres-Boran used LIBSVM to standardize a testing environment for studying different SVM kernel functions on the same set of data[46].

One area that SVMs excel is in the ability to classify between positive and negative connotations in text. Ceyhan and their team used an algorithm to train a Support Vector Classifier to group comments into two classifications, positive and negative. An SVC differs from an SVM in that the classifications are determined less on probability and more on context[47]. Bi utilized support vector machines to determine the orientation of customer reviews (positive vs. negative). The orientations were about four predetermined factors that affect a customer's perception of the service[40].

Another area in which SVMs excel is mathematical simplicity, but that can come at the cost of time. Chang implemented an SVM because it is best used for small sample sizes, high dimensional problems, and non-linear problems. SVMs follow a basic algorithm and are adaptable to many problems[5]. Senthurvelautham and Hettiarachchi believe that SVMs are time consuming due to the amount of computation required. However, when efficiently employed it can achieve the highest accuracy when finding aspects from customer reviews[48].

APPLICATION OF SUPPORT VECTOR MACHINES TO THE USPTO

Support Vector Machines pose the ability to extract the popular categories from a collection of text. These categories could include keywords related to customer sentiment and satisfaction. It could also be used to classify certain comments as being related to customer perceptions. Building a support vector machine for the purposes of this project may be unnecessary and open-source software may be pursued instead.



ASPECT-LEVEL ANALYSIS

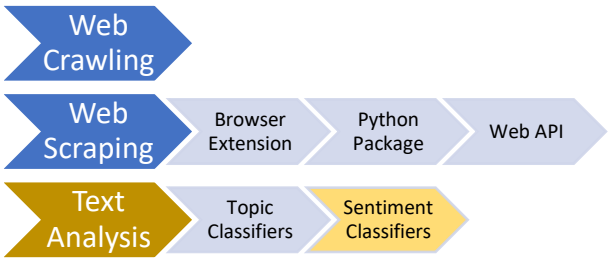
In the field of text analysis, there are three levels of depth that can establish sentiment: document-level, sentence-level and aspect-level. Aspect-Level Analysis (ALA) is the finest, most granular of the levels. An aspect can be described as a topic or sentiment within a clause of a sentence. This technique takes in documents of unstructured textual data. The analysis breaks down sentences into aspects, or attributes of the service, and determines the polarity of the sentiment. This creates a greater understanding of sentiment than other methods, because parts, or aspects, of the text are analyzed and categorized, rather than the whole group of text itself[49]. There are many ways of extracting aspects from sets of text. One way is to mine the most frequent nouns from the text set. In this field it is believed that the infrequent aspects of a document will be near to the frequent aspects. Therefore, by finding the frequent aspects first, the infrequent can be extracted. It is these infrequent aspects that can determine sentiment[50]. For example, a document discussing a trip to the beach. The beach may be the most frequent aspect of the document, but the specific parts of the trip are what drive the sentiment. If there were no bathrooms at the beach, bathrooms would be an infrequent topic that is closely associated with a frequent aspect. The fact that there were no bathrooms can be interpreted as negative sentiment.

Aspect level sentiment analysis is a close analysis of text and can be implemented in various ways. Aspect level sentiment analysis first extracts known aspects from each analyzed segment. Then, it determines the sentiment of each segment by associating the aspect with nearby sentiment related words[51]. Thet and their team collected clauses of sentences and broke the clauses into dependency

trees. These dependency trees were used to identify and remove words with less meaning. The trees were also used to improve the speed of searching through the data. These trees were then used to determine certain aspects of customer reviews related to movies on imbd.com[52]. Ali, Wang, and Riaz utilized aspect-based sentiment analysis on online reviews from a ride sharing company. By searching for the most frequent words, they were able to extract the most prevalent aspects of the text. They could then refine their search to just nouns, or another part of speech, as needed[53].

OPEN-SOURCE SOFTWARE FOR ALA

There are not many aspect-level sentiment analysis software applications available for public use. However, Haider and Nasim have developed software to accomplish such a task. This package allows the user to train an algorithm to respond to their data accurately, then they can deploy a web application that feeds the program comments for analysis. The analysis process considers term extraction, category detection, and polarity identification[54].



LINK ANALYSIS

Link analysis is used to find relationships between data. Links are found by identifying commonalities between two sets of data. Frequently, link analysis is used by law enforcement to understand the connections between criminal networks[55]. Many link analysis strategies were specifically used to quantify data and correlate relationships between trends, request, and actionable information.

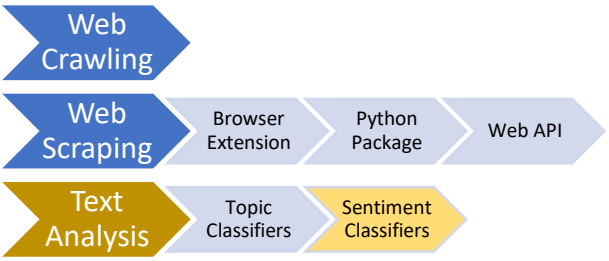
To conduct link analysis, multiple data sources are required. Some example inputs of data sources are events, websites, interviews, and people. The information gathered from each source must be recorded. For example, important information from an interview will not be collected unless the

interviewee's responses are recorded. Then, further analysis of information gathered is required. Finally, lines can be drawn between each source, or the information from each source[56]. When links are drawn, they can be based on similarity, differences, or trends.

Link analysis aggregates raw data to visualize relationships between the data[57]. According to Berezina et al, IBM's text-link analysis tool can be used to identify pairs of words used together frequently. They reported that refining the strategy of searching for pairings of companies with their products or services may be useful[58]. Cao used a link analysis tool on a transit customer blog page to quantify the riders' experiences with the train service. This tool was able to find highlights, setbacks and recommendations. Then, Cao was able to create an overall sentiment rating that the riders felt about the train service.

APPLICATIONS OF LINK-ANALYSIS TO THE USPTO

Link analysis at the USPTO can help identify relationships from various data sources. Posts that are made on Patently-O, IPwatchdog, Reddit, Twitter, and Facebook can all be collected and then link analysis can be applied to compare the trends, similarities, and differences of information from each source. For example, if there is a post about timeliness in which one person responds that the timeliness of the application process is too long, there will be multiple comments in response to this post. The entire chain of responses is called a conversation. Throughout this conversation there may be many forms of actionable information that can lead to deeper sentiment analysis. If a similar conversation were to happen on a different source, then the utilization of link analysis will allow a link to be drawn between the similar information from each source.



NAIVE BAYES CLASSIFIERS

A Naive Bayes Classifier is a classification algorithm that incorporates Bayes' Theorem[59]. Bayes' Theorem asserts the probability of an event, provided another specific event happened as well. When the algorithm is given training data, Bayes' Theorem is used to make a prediction to improve the algorithm's understanding of that data.

Bayes' Theorem in probability is used to test one category of data against another[60]. If data are analyzed with Bayes' Theorem, a Naive Bayes Classifier can be trained using that data to make categorical classifications[61]. The output probability of the Naive Bayes Classifier can be used to determine how likely a component of the data belongs in one category or another. This information is used to refine the algorithm's sense of what kinds of data are classified as belonging in each category.

Mushtaq and their team found ways to collect insight from various users. A Naive Bayes Classifier was used to investigate different perspectives of streaming delivery over cloud networks. The team investigated how different factors contribute to the Quality of Experience (QoE) for video streaming services. They employed a Naive Bayes Classifier to sort the factors into categories[62]. Grljevic and Bosnjak used a Naive Bayes Classifier to create categories to predict the classification of words. They found that when this was deployed over a dataset that considered the root of the word for connotation, it was 98% accurate[42].

Although a productive technique, Naive Bayes Classifiers can only be used in very specific circumstances. For example, Senthurvelautham and Hettiarachchi found in their research that although

Naive Bayes Classifiers have a high accuracy, they frequently have lower performance than Aspect-Level Analysis and Link Analysis[48].

Apache Spark is a machine learning tool that can use Naive Bayes Classifiers for predictions. It utilizes uses a network of other computers to compute complex problems[63]. Apache Spark can also be used for document classification, meaning it can categorize documents based on their text[64].

APPLICATION OF NAIVE BAYES CLASSIFIERS TO THE USPTO:

The USPTO could use a Naive Bayes Classifier to determine customer satisfaction in a specific area given a circumstance. Hypothetically, the USPTO could be interested in sorting through data to find only comments of negative sentiment related to a certain part of the application process. Naive Bayes Classifiers use Bayes' Theorem to analyze input data and cannot parse text on their own. To use a Naive Bayes Classifier, another tool would have to collect the data in advance of analysis.

EXPECTATION CONFIRMATION THEORY

When a customer is considering purchasing a product, they have an expectation of how they believe the product will perform. If the customer buys the product, the product will perform in a way that either matches the customer's expectation or does not. The study of this mindset is called Expectation Confirmation Theory (ECT)[65]. According to Tsao, Expectation Confirmation Theory has shown that satisfaction is a result of customer confirmation[66].

When considering Expectation Confirmation Theory, the difference between customer expectation and perceived performance is the area of interest[67]. Xu Xun employed Expectation Confirmation Theory as a framework for his customer feedback collection system that used Latent Semantic Analysis to determine the relevance of topics within online reviews. These reviews were seen to follow a phenomenon called the asymmetric effect. The asymmetric effect states that customer

feedback does not always completely reflect what a customer is feeling about a product or service. These results were compared against others with ECT in mind to understand the effect each review had as opposed to just what was said[30]. Machado utilized ECT to evaluate customer satisfaction with hotel services in Peru. They analyzed survey inquiries, and considered customer reported satisfaction and hotel booking prices for context. Ultimately, ECT held true, as hotel features were determined to have a direct impact on customer satisfaction[68].

The USPTO can use ECT to interpret the feedback it receives and measure the feedback against customer perception goals. In addition to gauging this feedback, the USPTO may want to consider using ECT to further understand the perspective of QoE from the customer point of view. For instance, Cao and Cao needed a way to understand if the riders of the various transit systems were satisfied with the service. Throughout the report, Cao explains why the experience of the transit service varies. Had the transit service used Expectation Confirmation Theory, customers' expectations could have been compared with their experiences. Using the same theory, the transit service would be able to rule out outliers like those with expectations that are too high[3].

EVALUATING PASSIVE DATA TECHNIQUES

The team's research showed significant advantages in using PDC for gathering customer perceptions. Based on studies done by technicians, industry professionals, and scholars, the use of PDC methods are promising for gathering actionable information. At the USPTO, PDC will lead to a higher level of sentiment analysis. Conducting research into several collection, analysis, and ethical elements of PDC lead to the need for an evaluation matrix. The team was able to narrow our search to six best practice PDC analysis methods. The OPQA Professionals at the USPTO will be able to review our matrix to then determine the best fit method for the type of PDC the office would like to conduct for gathering actionable information.

[1] What is Web Scraping and How Does Web Crawling Work? (n.d.). *Scrapinghub*. Retrieved November 16, 2020, from https://www.scrapinghub.com/what-is-web-scraping/

[2] Guo, Y., Barnes, S. J., & Jia, Q. (2017). Mining meaning from online ratings and reviews: Tourist satisfaction analysis using latent dirichlet allocation. *Tourism Management*, *59*, 467–483. https://doi.org/10.1016/j.tourman.2016.09.009

[3] Cao, J., & Cao, X. (2017). Comparing importance-performance analysis and three-factor theory in assessing rider satisfaction with transit. *Journal of Transport and Land Use*, *10*(1). https://doi.org/10.5198/jtlu.2017.907

[4] Lyu, F., & Choi, J. (2020). The Forecasting Sales Volume and Satisfaction of Organic Products through Text Mining on Web Customer Reviews. *Sustainability*, *12*(11), 4383. https://doi.org/10.3390/su12114383

[5] Chang, A.-C., Trappey, C., Trappey, A. J. C., & Chen, L. W. L. (2020). Web Mining Customer Perceptions to Define Product Positions and Design Preferences. *International Journal on Semantic Web and Information Systems*, *16*(2), 42–58. https://doi.org/10.4018/IJSWIS.2020040103

[6] Matta, P., Sharma, N., Sharma, D., Pant, B., & Sharma, S. (2020). Web scraping: Applications and scraping tools. *International Journal of Advanced Trends in Computer Science and Engineering*, *9*(5), 8202–8206. Scopus. https://doi.org/10.30534/ijatcse/2020/185952020

[7] *What is Web Scraping and What is it Used For? | ParseHub*. (2019, August 6). ParseHub Blog. https://www.parsehub.com/blog/what-is-web-scraping/

[8] *Introduction to web scraping: Manually scrape data using browser extensions*. (n.d.). Retrieved November 20, 2020, from https://librarycarpentry.org/lc-webscraping/03-manual-scraping/index.html

[9] *What are extensions?* (n.d.). MDN Web Docs. Retrieved November 23, 2020, from https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/What_are_WebExtensions

[10] About Us | GitHub. (n.d.). Retrieved December 09, 2020, from https://github.com/about

[11] PyPI · The Python Package Index. (2020). PyPI. Retrieved December 4, 2020, from https://pypi.org/

[12] *Beautiful Soup Documentation—Beautiful Soup 4.9.0 documentation*. (n.d.). Retrieved November 24, 2020, from https://www.crummy.com/software/BeautifulSoup/bs4/doc/

[13] *Selectors—Scrapy 2.4.1 documentation*. (n.d.). Retrieved November 24, 2020, from https://docs.scrapy.org/en/latest/topics/selectors.html#topics-selectors

[14] Landers, R.N., Brusso, R.C., Cavanaugh, K.J., & Collmus, A.B. (2016). A primer on theory-driven web scraping: Automatic extraction of big data from the internet for use in psychological research. *Psychological Methods*, *21*(4), 475–492. Scopus. https://doi.org/10.1037/met0000081

[15] Beal, V. (n.d.). *What is Web Scraping? Webopedia Definition*. Retrieved November 4, 2020, from https://www.webopedia.com/TERM/W/Web_Scraping.html

[16] Budiarti, R. P. N., Widyatmoko, N., Hariadi, M., & Purnomo, M. H. (2016). Web scraping for automated water quality monitoring system: A case study of PDAM Surabaya. *2016 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, 641–648. https://doi.org/10.1109/ISITIA.2016.7828735

[17] Diouf, R., Sarr, E. N., Sall, O., Birregah, B., Bousso, M., & Mbaye, S. N. (2019). Web Scraping: State-of-the-Art and Areas of Application. *2019 IEEE International Conference on Big Data (Big Data)*, 6040–6042. https://doi.org/10.1109/BigData47090.2019.9005594

[18] *The Impacts of Service Failure and Recovery Efforts on Airline Customers' Emotions and Satisfaction—Xun Xu, Wenhui Liu, Dogan Gursoy, 2019*. (2018). Retrieved October 27, 2020, from https://journals.sagepub.com/doi/full/10.1177/0047287518789285

[19] Twitter API Documentation | Docs | Twitter Developer. (n.d.). Retrieved December 09, 2020, from https://developer.twitter.com/en/docs/twitter-api

[20] Introduction to web APIs. (n.d.). Retrieved December 09, 2020, from https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Introduction

[21] Peacock, S. E. (2014). How web tracking changes user agency in the age of Big Data: The used user. *Big Data & Society*, *1*(2), 2053951714564228. https://doi.org/10.1177/2053951714564228

[22] Taylor, I. (2017). Data collection, counterterrorism and the right to privacy. *Politics, Philosophy & Economics*, *16*(3), 326–346. https://doi.org/10.1177/1470594X17715249

[23] Nunan, D., & Di Domenico, M. (2013). Market Research and the Ethics of Big Data. *International Journal of Market Research*, *55*(4), 505–520. https://doi.org/10.2501/IJMR-2013-015.

[24] Younes, A. S. (2019). Passive violation of consumers' privacy rights on the internet in the age of emerging data capital. *Journal of Content, Community and Communication*, *10*(5), 134–150. Scopus. https://doi.org/10.31620/JCCC.12.19/14

[25] Clifton, C. (n.d.). *Data mining | computer science*. Encyclopedia Britannica. Retrieved November 15, 2020, from https://www.britannica.com/technology/data-mining

[26] Alexandra Twin. (n.d.). *Data Mining: How Companies Use Data to Find Useful Patterns and Trends*. Investopedia. Retrieved November 3, 2020, from https://www.investopedia.com/terms/d/datamining.asp

[27] *What is Text Mining, Text Analytics and Natural Language Processing?* (2019, August 28). Linguamatics. https://www.linguamatics.com/what-text-mining-text-analytics-and-natural-language-processing

[28] Liao, X., Chen, G., Ku, B., Narula, R., & Duncan, J. (2020). Text Mining Methods Applied to Insurance Company Customer Calls: A Case Study. *North American Actuarial Journal*, *24*(1), 153–163. https://doi.org/10.1080/10920277.2019.1649155

[29] Liu, J., Zhou, Y., Jiang, X., & Zhang, W. (2020). Consumers' satisfaction factors mining and sentiment analysis of B2C online pharmacy reviews. Bmc Medical Informatics and Decision Making, 20(1), 194. https://doi.org/10.1186/s12911-020-01214-x

[30] Xu, Xun. (2018). Does traveler satisfaction differ in various travel group compositions? *International Journal of Contemporary Hospitality Management. 30*(3). https://doi.org/10.1108/IJCHM-03-2017-0171

[31] Yoon, B. et al. (2020) A systematic approach to prioritizing R&D projects based on customer-perceived value using opinion mining. *Technovation. 98.* https://doi.org/10.1016/j.technovation.2020.102164

[32] Trappey, A. J. C. et al. (2017). Mining the Customer's Voice and Patent Data for Strategic Product Quality Function Deployment. *Transdisciplinary Engineering: A Paradigm Shift*. *5* pp. 985–992 https://doi.org/10.3233/978-1-61499-779-5-985

[33] Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, *25*(2–3), 259–284. https://doi.org/10.1080/01638539809545028

[34] *Fit LSA model—MATLAB fitlsa*. (n.d.). Retrieved November 17, 2020, from https://www.mathworks.com/help/textanalytics/ref/fitlsa.html

[35] Welcome to PLSA's documentation. (n.d.). Retrieved December 09, 2020, from https://probabilistic-latent-semantic-analysis.readthedocs.io/en/latest/index.html

[36] X. Xu, W. Liu & D. Gursoy. (2018). The Impacts of Service Failure and Recovery Efforts on Airline Customers' Emotions and Satisfaction. *Sage Journals, 58*(6). https://doi.org/10.1177%2F0047287518789285.

[37] Marcolin, C. et al. (2019). Business analytics in tourism: Uncovering knowledge from crowds. *Brazilian Administrative Review. 16*(2). https://doi.org/10.1590/1807-7692bar2019180136

[38] Blei, D. M. (2003). Latent Dirichlet Allocation. *Journal on Machine Learning Research 3*(4-5). From, https://www.scopus.com/record/display.uri?eid=2-s2.0-0141607824&origin=resultslist&sort=plf-f&src=s&st1=latent+dirichlet+allocation&nlo=&nlr=&nls=&sid=a6106ecb88598dabefdca15a08d11db0&sot=b&sdt=cl&cluster=scopubyr%2c%222003%22%2ct&sl=42&s=TITLE-ABS-KEY%28latent+dirichlet+allocation%29&ref=%28blei%29&relpos=1&citeCnt=19059&searchTerm=

[39] Lock, O., Pettit, C. (2020). Social media as passive geo-participation in transportation planning. *Geo-Spatial Information Science*. https://doi.org/10.1080/10095020.2020.1815596

[40] Bi, J.-W., Liu, Y., Fan, Z.-P., & Cambria, E. (2019). Modelling customer satisfaction from online reviews using ensemble neural network and effect-based Kano model. *International Journal of Production Research*, *57*(22), 7068–7088. https://doi.org/10.1080/00207543.2019.1574989

[41] Joachims, T. (1998). Text categorization with Support Vector Machines: Learning with many relevant features. In C. Nédellec & C. Rouveirol (Eds.), *Machine Learning: ECML-98* (Vol. 1398, pp. 137–142). Springer Berlin Heidelberg. https://doi.org/10.1007/BFb0026683

[42] Grljevic, O., & Bosnjak, Z. (2018). Sentiment Analysis of Customer Data. *Strategic Management*, *23*(3), 38–49. https://doi.org/10.5937/StraMan1803038G

[43] Guan, F., Shi, J., Cui, W., Hong, D., & Wu, J. (2019). A method for false alarm recognition considering threshold. *2019 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC)*, 1043–1049. https://doi.org/10.1109/SDPC.2019.00199

[44] *An Introduction to Support Vector Machines (SVM)*. (2017, June 22). MonkeyLearn Blog. https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/

[45] *Module: Tf | TensorFlow Core v2.3.0*. (n.d.). Retrieved November 24, 2020, from https://www.tensorflow.org/api_docs/python/tf

[46] Torres-Barrán, A., Alaíz, C. M., & Dorronsoro, J. R. (2021). Faster SVM training via conjugate SMO. *Pattern Recognition*, *111*, 107644. https://doi.org/10.1016/j.patcog.2020.107644

[47] Ceyhan, M., Orhan, Z., Karras, D. A., & Dane, S. (2020). Sentiment Analysis of Hospital Service Satisfaction. *Journal of Research in Medical and Dental Science*, *8*(5), 7.

[48] Senthurvelautham, S., & Hettiarachchi, S. (2018). SentScore: Autonomous Text Sentiment scoring and Summarizing System related to Complaint Management. *2018 4th International Conference for Convergence in Technology (I2CT)*, 1–8. https://doi.org/10.1109/I2CT42659.2018.9058199

[49] *A Comprehensive Guide to Aspect-based Sentiment Analysis*. (2019, March 8). MonkeyLearn Blog. https://monkeylearn.com/blog/aspect-based-sentiment-analysis/

[50] Farhadloo, M. (2015). *Statistical Models for Aspect-Level Sentiment Analysis* [UC Merced]. https://escholarship.org/uc/item/2ks913br

[51] Trigui et al. (2020). ADAL System: Aspect Detection for Arabic Language. *Hybrid Intelligent Systems 1179*. https://link-springer-com.ezpxy-web-p-u01.wpi.edu/chapter/10.1007%2F978-3-030-49336-3_4

[52] Thet, T. T., Na, J.-C., & Khoo, C. S. G. (2010). Aspect-based sentiment analysis of movie reviews on discussion boards. *Journal of Information Science*, *36*(6), 823–848. https://doi.org/10.1177/0165551510388123

[53] Ali, S., Wang, G., & Riaz, S. (2020). Aspect Based Sentiment Analysis of Ridesharing Platform Reviews for Kansei Engineering. *Ieee Access*, *8*, 173186–173196. https://doi.org/10.1109/ACCESS.2020.3025823

[54] Nasim, Z., & Haider, S. (2017). ABSA Toolkit: An Open-Source Tool for Aspect Based Sentiment Analysis. *International Journal on Artificial Intelligence Tools*, *26*. https://doi.org/10.1142/S0218213017500233

[55] 2019-07-15 |. (n.d.). *Link analysis: The lynchpin to better investigations*. Retrieved November 12, 2020, from https://www.visallo.com/blog/link-analysis-better-investigations/

[56] Disney, A. (2020, January 30). *Link analysis for fraud detection: A step-by-step example*. Cambridge Intelligence. https://cambridge-intelligence.com/link-analysis-fraud-detection/

[57] Si, Y., Cheong, S., Fong, S., Biuk-Aghai, R. P., & Cheong, T. (2012). A layered approach to link analysis and visualization of event data. *Seventh International Conference on Digital Information Management (ICDIM 2012)*, 181–185. https://doi.org/10.1109/ICDIM.2012.6360101

[58] Berezina, K., Bilgihan, A., Cobanoglu, C., & Okumus, F. (2016). Understanding Satisfied and Dissatisfied Hotel Customers: Text Mining of Online Hotel Reviews. *Journal of Hospitality Marketing & Management*, *25*(1), 1–24. https://doi.org/10.1080/19368623.2015.983631

[59] Gandhi, R. (2018, May 17). *Naive Bayes Classifier*. Medium. https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c

[60] Chauhan, G. (2018, October 8). *All about Naive Bayes*. Medium. https://towardsdatascience.com/all-about-naive-bayes-8e13cef044cf

[61] Josephine Theresa, S., & Evangeline, D. J. (2021). Classification of Diabetes Milletus Using Naive Bayes Algorithm. In J. D. Peter, S. L. Fernandes, & A. H. Alavi (Eds.), *Intelligence in Big Data Technologies—Beyond the Hype* (pp. 401–412). Springer. https://doi.org/10.1007/978-981-15-5285-4_40

[62] Mushtaq, M. S., Augustin, B., & Mellouk, A. (2012). *Empirical study based on machine learning approach to assess the QoS/QoE correlation* (p. 7). https://doi.org/10.1109/NOC.2012.6249939

[63] *10 Most Popular Machine Learning Software Tools in 2020 (updated) | by Sophia Martin | Towards Data Science*. (n.d.). Retrieved November 15, 2020, from https://towardsdatascience.com/10-most-popular-machine-learning-software-tools-in-2019-678b80643ceb

[64] *Naive Bayes—RDD-based API - Spark 3.0.1 Documentation*. (n.d.). Retrieved November 22, 2020, from https://spark.apache.org/docs/latest/mllib-naive-bayes.html

[65] *Nevo, D. (2015). Expectation confirmation theory. https://is.theorizeit.org/wiki/Expectation_confirmation_theory.*

[66] Tsao, W.-Y. (2013). Application of Expectation Confirmation Theory to Consumers' Impulsive Purchase Behavior for Products Promoted by Showgirls in Exhibits. *Journal of Promotion Management*, *19*(3), 283–298. https://doi.org/10.1080/10496491.2013.770811

[67] Elkhani, N., & Bakri, A. B. (n.d.). *REVIEW ON "EXPECTANCY DISCONFIRMATION THEORY" (EDT) MODEL IN B2C E-COMMERCE*. 13.

[68] Machado, M. J. C. V. (2019). Determinants of customer satisfaction: Empirical study in hotels. *International Journal of Applied Management Science*, *11*(2), 91–112. Scopus. https://doi.org/10.1504/IJAMS.2019.098823