Initial Solution Development for Text Data Analysis at InterDesign

By

Salil Gupta

A Capstone Project Paper Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

In

Data Science

University of Wisconsin – Eau Claire

Eau Claire, Wisconsin

July 2019

ABSTRACT

Initial Solution Development for Text Data Analysis at InterDesign

Salil Gupta

Capstone Project for Data Science Master’s Degree

University of Wisconsin, Eau Claire

Eau Claire, WI

July 2019

At InterDesign, new business focus on Digital Marketing, Social Media, and Brand presence, along with, the availability of massive amounts of data, has created a demand for new technological solutions that will allow employees to excel at driving growth.

With a large amount of the information being unstructured text, and without a strategy to do meaningful text analysis, the Business Intelligence team has initiated development of a prototype Data Science process that allows the organization to better understand the words that people associated with the brand are saying. The final product of the initial development is an easy to understand, versatile set of data load, storage, analysis and reporting processes that can be customized and further developed to provide the organization with continuous meaningful insights.

Table of Contents

Introduction…………………………………………………………………………....……….……1

Role of the Business Intelligence Team…………………………………...………….……1

Consumer Marketing Strategy………………………………………………...………………2

Interdesign Social Currency Examples……………………………………….……………4

Measuring Success in Marketing Initiatives………………………………….….….……6

PROBLEM AND SolUTION Identification………………………..…………….……….….……7

Solution Development Methodology………….…………………..….…………….....……8

Provisioning Resources……………………………………………………….…………………..9

SETTING UP CLOUD ENVIRONMENT……………………………………..……..…..………………10

Accessing Twitter Data…………………………………………………………….……………16

Extract, Transform, and Load………………………………………..………………………19

DATA STORAGE…………………………………………………………….……..……………………22

DATA ANALYSIS………………………………………………………………………………..………23  
REPORTING……………………………………………………………….………………..……………41

Report USAGE…………………………………………………………………………………………42  
CONCLUSION OF DEVELOPMENT………………………..…………………………….……………42  
NEXT STEPS……………………………………………………………………………..………………43  
REFERENCeS………………………………………………….…………………………………………45

LISt OF FIGURES

Figure 1: Embrace your Space July 5 School Supply Tweet……………………...………………4

Figure 2: Nikki Boyd Plastic Pouches Native Advertising ……………………….….……...……5

Figure 3: Azure Main Screen………………………………………………….….…..….………11

Figure 4: Add Resource Group……….…………………………...……………….….…………12

Figure 5: Associate Resource Group with Subscription………………………..….….…………12

Figure 6: Add Virtual Machine…………………………………………………….….…………13

Figure 7: Virtual Machine Specification Input…………………………………….…….………14

Figure 8: Allow RDP Port to Virtual Machine…………………………………….…….………15

Figure 9: Connect and Download Remote Desktop Connection………………….……..………15

Figure 10: Resize Virtual Machine Options…………………………………….……….………16

Figure 11: Twitter API Access Chart………………………………………….………...….……17

Figure 12: Register Application……………………………………………….….….…..………18

Figure 13: Generate Authorization Credentials……………………………….……..….…….…18

Figure 14: Virtual Machine CPU Usage…………………………………….……….…..………21

Figure 15: ETL Sleeping between Paginated Requests…………………….………..…..………22

Figure 16: Recent tweets over time………………………………………….………..…………26

Figure 17: Mentions of the word ‘DIY’ over time………………………….………...….……...28

Figure 18: Wordcloud……………………………………………………...………….…………29

Figure 19: Sentiment of recent tweets…………………………………….………….….………32

Figure 20: Word Contribution to Sentiment Polarity…………………….…………….…..……33

Figure 21: Sentiment Comparison Cloud……………………………………………….….……34

Figure 22: Printed Nested Time Series………………………………………………….…….…36

Figure 23: Printed Nested Vector Item…………………………………………………….….…37

Figure 24: Terms with large change in mention rate………………………...………………..…39

Figure 25: Most mentioned 3-word phrases……………………………………..………………40

Figure 26: Bigram Network……………….……………………………………..………………41

List of TABLES

Table 1: Sample Record………………………………………………………………...………21

Table 2: Input R Data Structure …………………………………………………………..……25

Table 3: Tidy Words Data Structure……………………………………………………………25

Table 4: Sentiment Join Result Data Structure……….………………...………………………30

Table 5: Words Over Time Data Structure………………………..……………………………36

Table 6: Nested Time Series Data Structure……………………………………………………36

Table 7: Nested Models Data Structure ………………………..………………………………38

Table 8: Unnested Models Data Structure ………………………………………..……………38

Introduction

InterDesign is a houseware design company with product solutions designed for the bathroom, kitchen, pantry, closet, laundry room and other areas of the home.

Innovating the product solutions are a team of in-house engineers and designers that work to create a product line that is as functional as is beautiful. InterDesign prides itself on igniting homemakers’ creativity with fresh ideas on how to revitalize their home and, in turn, the organization listens to the voice of its consumers, anticipates their needs and exceeds their expectations. Furthermore, the organization looks to gain inspiration from organization advocates, and Do-it-Yourself experts.

As a global leader in its retail space, the organization has customers in over 100 countries across several continents. InterDesign has traditionally sold products in bulk straight to distributors, however, the organization has had recent success with a straight to consumer business model selling products directly through online marketplaces.

With the increased growth of this business, leadership at the organization has made marketing products to everyday consumers one of its top priorities.

Role of the Business Intelligence Team

As a large e-commerce and retail organization there are many simple and complex data analysis problems. The extent of these problems necessitates a more effective approach to data usability. For this reason, the organization believes data should be available not just to core business groups that use it, but data that provides context should be available in its clearest form to everyone in the organization. Whenever useful and intelligent data is not made widely available, an organization risks a business process relying on subpar information because employees aren’t aware of a better information source.

The Business Intelligence (BI) Developer Team’s function is to identify and provide this type of access to the data, and to prevent business groups from relying on second-tier information. As many departments frequently use a large amount of data, it is important for the team to identify important systems, spreadsheets, and other data sources. Additionally, the team, and its developers, should understand how the organization will load, store, and analyze it. With the organization transitioning into the e-commerce business, there are multiple sources that will require new technologies for it to be able to integrate the data into its core business processes.

If the team doesn’t have a legitimate strategy for integrating new technologies to extract useful information, the organization runs the risk of departments buying ineffective and costly solutions due to a lack of adequate internal alternatives.

Consumer Marketing Strategy

The organization has strived to maintain strong relationships with its consumers wherever it interacts with them, and traditionally, this has been large retailers who help distribute InterDesign’s product lines. However, with the increasing success of the straight to consumer e-commerce business, it is even more important that the organization focuses on creating and maintaining its presence as a household name.

The core brand of InterDesign is iDesign, often displayed in tandem with its slogan of “Live Simply.” iDesign is used to market products on a number of marketing channels including the website, Instagram, Facebook, and Twitter.

In order to elevate iDesign to a household name the organization has determined it is necessary to:

* Gather the right information about its niche
* Focus on Social Media
* Master Native Advertising

In general, the organization has decided to focus its efforts on digital marketing over traditional marketing because of the common advantages it provides:

* Global Reach: Web and Social Media channels have the potential for global reach. The organization can maintain brand advocates who have a predominant following of international customers.
* Cost Effectiveness: Digital Marketing is generally more cost effective because in the long run you can reach the right customers at a lower cost than traditional marketing.
* Trackable Measurable Results: Digital Marketing interactions are easier to track than traditional interactions since important details are digitally stored. Additionally, analytical tools for the web are widely available, and, often, easy to use.
* Improved Conversation Rate: With digital marketing methods, customers are only a few clicks away from completing a purchase. Offline advertising methods require customers to take additional steps to make a purchase. For example, they might have to drive to a store, type in a URL, or find a hyperlink.
* Personalization: Digital Marketing Channels can be more personalized. Websites can be designed to have showcased products and targeted offers based on browser history. Marketing through social media can be done in a personalized way by following, friending, subscribing, etc.
* Social Currency: Digital Marketing campaigns build social currency. Social currency is a common concept in marketing and business these days that refers to the pull or influence that a customer has among his or her peers, and social currency strategies are arguably the most critical point of leverage for a marketer.

In other words, Social currency is the measured value of word of mouth. Because customers effectively interact with human beings, commercial transactions take on social value. Relevance and regard create a store of value of social currency that these products are responsible for. (Klein, 2018)

The modern media landscape has fostered an environment with real time connectivity and has transformed old fashioned product announcements into ongoing back and forth conversations.

Whereas paid traditional advertising postings are so obvious, postings by twitter friends are more organic.

Furthermore, when brand advocates generate traffic through their organic posts, they could have a more lasting positive impact on followers. A post generated today could get retweeted and liked months, or even years, later. When brand advocates start a conversation about something, that conversation has the potential to continue indefinitely. They can keep posting and followers can refer to their profiles for as long as they are interested.

In this way, brand advocates themselves can be considered as long—term assets. They represent brand presence, and social currency, and ultimately, they bring a whole set of social values that is associated with the digital content they publish.

InterDesign Social Currency ExaMPLES

The following tweet is an example of the Social Currency that InterDesign has generated. This post by Embrace Your Space, @EmbraceSpaceNYC, highlights the organization capability of drawer dividers, while also driving social value in the concept of organization, responsible parenting, and education.A screen shot of a book shelf

Description automatically generated

Figure : Embrace your Space July 5 School Supply Tweet

Another example, is this post by Nikki Boyd which does a good job of showcasing the versatility, and functionality of InterDesign packing pouches. It, originally, was one tweet by Nikki Boyd, through her twitter handle, @homewithnikki, with a link to a YouTube video. However, later, the twitter post was retweeted, and liked, and people commented back and forth on the YouTube video.

A screenshot of a cell phone

Description automatically generated

Figure : Nikki Boyd Plastic Pouches Native Advertising

Measuring Success in Marketing Initiatives

With the increased importance placed on new marketing methods, social media and brand presence, it is also increasingly important the organization get a detailed measure of its incremental success. This entails evaluating Market Channel Return On Investment(ROI), growth in brand presence, and growth in social currency.

Accurately measuring the ROI across the different marketing channels is not a simple task. It requires tracing back sales transactions to the marketing channel that was most effective in converting.

For instance, when a customer purchases one of InterDesign’s products in a retail store, it is difficult to figure out if that customer had seen one of the organization’s advertising campaigns.

Unfortunately, IT doesn’t have the strategy or skills to solve this problem that has always traditionally existed. However, with the focus moving to digital marketing, there has been an increased demand for internal IT data analysis skills.

Digital Marketing that generate digital sales are easier to measure since you have trackable data points at all steps of the customer journey.

For instance, when someone goes to the website and looks at a product, they can follow a link to a partner’s page where they can order it online. In this way downstream revenue is linked to the specific ad channel, the website. Additionally, through common web analytic methods, the organization has methods for understanding the consumer’s digital footprint and consumer behavior up to the point of purchase. They can answer complex analytical questions that help understand ROI.

These are questions like:

* Did the purchasing customer visit many pages on the site?
* Did he purchasing customer stay at the main screen for a long or short time?
* Is the purchasing customer a repeat visitor, or a new visitor?
* If a website user is on a page, or clicks a link, what is the likelihood they will make a purchase?
* What is the average amount spent by customers with specific web activity?

Answers to these questions provides the business with a very good picture of website ROI.

Measuring the return on investment of social media is much less straight forward. Furthermore, evaluating brand presence on social media, and the social currency that social media generates, is even more complicated.

This had led the organization to ask the questions:

* How can it begin to develop processes that measure growth in social currency?
* How does it identify those specific social values that its customers respond to?
* How does it identify the social values that its influencers represent?

By understanding this information, and identifying the more effective social values, the organization can guide itself on which influencers to champion its brand.

More importantly if the organization doesn’t truly understand what values its influencers represent, then the same asset that generates revenue could become a liability when consumers associate them with the wrong principles. So how does the organization detect when it has a loss in social currency because a brand advocate is suddenly associated with values contrary to the business?

The organization doesn’t have any methods for understanding the large volume of digital content that customers, and brand advocates, who “socialize” with social media accounts, will share with the rest of the world, so it is imperative the Business Intelligence team develops a strategy to begin to understand what content they publish.

PROBLEM AND SolUTION Identification

Most of the social media data that exists is inherently, text, images, and video. While there are big data analysis methods for images, and video, the team of Business Intelligence developers will be focusing on the most solvable problem, and one in which the majority of info resides: text. For the same reason, because it’s a very solvable problem, and because there is a large volume of info, the BI team will be focusing on Twitter data.

On Twitter most of the content that people post are the texts in tweets. Reading the text itself gives someone a good indicator of the emotions and social values the authors want to convey.

However, InterDesign generates a lot of text, and social currency, through the Twitter Account. The Twitter handle, @iDLiveSimply, is very active in publishing new tweets, as well as following people and organizations associated with organized living, do-it-your-self experts, and homemakers, and attracting new followers, retweets, and likes. The organization is following 353 twitter handles, has 356 followers and has 1068 tweets and 1197 likes.

Because of this, it’s too costly to have people constantly scan Twitter accounts, the organization needs a way to automatically understand the sentiments and ideas in those texts. Additionally, text data exists in a number of other sources, like customer feedback.

Since text analysis has proven methods of reliably understanding massive amounts of text data, the organization wants to begin exploring and understand some of these methods to aid consumer marketing.

This is a useful problem for the IT Business Intelligence team to focus on because it demonstrates progress in several important long-term initiatives within the organization:

* Assist Marketing with Making InterDesign a Household Name
* Make new information more widely available
* Show Progress on understanding Digital Content
* Show Progress on Developing Text Analytic Methods

Solution Development Methodology

For the initial Proof of Concept development the Business Intelligence team of developers will look to create a data science solution that accesses, loads, stores, and analyzes the digital content published by social media influencers in Twitter timelines so the organization can get a better understanding of the social currency they represent.

To develop sustainable processes, the team will adhere to certain principles:

* Showing incremental value in analysis: Since the organization is just starting to get into text analysis the development team will want to create processes that show incremental progress in the insight it draws from text. Furthermore, the team wants to develop methods, code, and processes that will be used for similar data sources down the road.
* Show learning and understanding of new methods: The team needs to learn more about text analytics and show a deep understanding of the methods it develops.
* Allowing Internal Data Access: Whenever possible, the BI team will look to make useful data available to anyone within the organization who can utilize it.
* Leverage Data Warehousing Skills and Concepts: Since the team of BI developers are already well-skilled in SQL, it will look to leverage its strengths in SQL Syntax, and SQL Server whenever it can.
* Stay Cost Effective
* Develop New Processes with Existing Infrastructures and Processes in Mind
* Support Existing Partners
* Development and Test in a limited, isolated, Sandbox Environment

Provisioning Resources

For any data analysis process, the BI team needs to ensure they have compute and storage power allocated for the analysis. This could be their local machine they are working on, a server in their local network, or some other virtual service obtained through the internet.

In this case, the team is utilizing cloud compute and storage resources for this data process because it provides a lot of advantages over traditional on-premise computing and storage.

* Scarcity of On-Premise Resources – Within the organization existing on-premise resources are scarce, as new machines are ordered when needed in production and utilized within a department to serve its purpose for its lifetime.
* Faster, Easier Setup - Setting up new hardware is a labor, and time intensive process where specifications are determined, quotes are priced out, hardware is ordered, received, and installed. It requires many department leaders to sign off on technical and usage details.
* Cost Efficiency of Cloud– Cloud computing can be cost efficient in use. In Microsoft Azure, a user only pays for compute when they need to use it. Instead of investing hundreds or thousands of dollars on hardware and software licensing fees, the organization can use cloud services to test the water with processes that use limited compute for a short amount of time.
* Scalability – Cloud computing is scalable as using more resources doesn’t require additional hardware components or licensing fees. In a traditional On-Premise Relational Database Environment, scaling up often means migrating data from a legacy machine, to a newer expensive machine.
* Backup and Recovery – Data and Applications hosted in the cloud are more easily able to the backed up and restored. Additionally, it is less prone to disaster or damage since data centers are physically optimized to prevent data loss.
* Ease of Deployment – Cloud processes can be deployed in different environments very easily because of the portability of cloud resources. Cloud resources can easily be cloned or migrated.

SETTING UP CLOUD ENVIRONMENT

The organization will be utilizing Microsoft Cloud’s product, Azure. In order to utilize compute and storage resources on Azure, the BI developer team needs to ultimately provision Virtual Infrastructure, under the Infrastructure-as-a-Service model.

In this model, the team is essentially setting up virtualized computing infrastructure. An alternative model for data processing, Platform-as-a Service, would mean the team is utilizing Compute for data processing through a specific service, bypassing virtualized infrastructure. For the initial solution development the Virtual Infrastructure will be a Virtual Machine.

However, before the team, or any new user, can set up a virtual machine, they need to do administration including linking a Microsoft Account and setting up their Subscription.

* Access to Azure requires an existing, or newly created, Microsoft Account. To set this up the user goes to <https://account.microsoft.com/account?lang=en-us>
* Once the user has a Microsoft account they can go to <https://portal.azure.com/>
* If a user hasn’t setup an Azure subscription, this page will ask them to set up an Azure Subscription.
* Azure Subscriptions is how Microsoft manages payment for the service. To create an Azure subscription, a potential subscriber needs to provide an online payment system, or debit or credit card.

Once the subscription is set up, a user can see all the different services available on the toolbar the left.

A screenshot of a cell phone

Description automatically generated

Figure : Azure Main Screen

Using the left-hand side toolbar, the user needs to select ‘Resource Groups’, to open the window to Add an Azure Resource Group.

A screenshot of a cell phone screen with text

Description automatically generated

Figure : Add Resource Group

* Azure Resource Groups are how resources are managed.
* Azure Resource Groups are associated with Azure Subscriptions to make it effective in managing the cost of different groups of resources.
* A user can’t utilize any Azure service without associating it with an Azure Resource Group, in turn associating it with an Azure Subscription. In this way a user can manage what account pays for what resource.

A screenshot of a cell phone

Description automatically generated

Figure : Associate Resource Group with Subscription

Now that the Resource Group and Subscription is setup, the user can ‘Add’ a Virtual machine, after navigating to Virtual Machine using the left-hand side toolbar.

A screenshot of a cell phone screen with text

Description automatically generated

Figure : Add Virtual Machine

This will take the user to a window where they can provision their Machine.

Provisioning a machine requires 6 steps:

1. Name the virtual machine. This will be the machine name once a user connects to it.
2. Select the region. This will direct the user to which region the datacenter where their Virtual Machine’s data resides. The advantage of selecting a region near the organization, is latency is reduced while onsite users are accessing it. The advantages of selecting different regions, potentially far away, is that the organization might be protected from local geographic disasters in a specific region.
3. Select the availability options. Here there are 3 options. A user can select no infrastructure redundancy. Or a user can select an Availability Set, which means that there will be a cluster of machines in the same data center supporting the VM service. Or a user can select an Availability Zone, which means that there will be a cluster of machines in different data centers in the same region supporting the VM Service.
4. Select the Image to be on the Virtual Machine. Inherently, the user is selecting the Operating System.
5. Select the VM “Size.” Although Size may sound like how much permanent storage space is on the disk, the user is selecting the CPU and memory. Azure Virtual Machines come in General Purpose Sizes that provide Balanced CPU-to-memory ratio. It’s ideal for small to medium databases, and low to medium traffic web servers. The VM size used in the process is A2Mv2. According to Microsoft, the Av2 series have CPU performance and memory configurations best suited for entry level workloads like development and test. The size is throttled, based upon the hardware, to offer consistent processor performance for the running instance, regardless of the hardware it is deployed on.
6. Set up access credentials, or Administrator username and password. This step is necessary anytime someone installs an operating system of a new machine.

A screenshot of a cell phone

Description automatically generated

Figure : Virtual Machine Specification Input

An additional step that is necessary for a user if they want to remote into their newly created machine, which the BI team does, is to Allow Inbound Port Rules so they can work with the virtual machine through Remote Desktop Application.

A screenshot of a cell phone

Description automatically generated

Figure : Allow RDP Port to Virtual Machine

Now the user’s Virtual Machine is ready to be started, and connected to via Remote Desktop Application(RDP). The user can download the file used to connect to RDP through the Virtual Machine tab in the portal. The user can use it like any other Windows machine at this point to run their data analysis software.

A screenshot of a cell phone screen with text

Description automatically generated

Figure : Connect and Download Remote Desktop Connection

If these resources are not adequate, the user has the flexibility of adding more resources without migrating the code or data from the machine. They can “resize” the machine from the original A2Mv2 to multiple, different options that are available.

A screen shot of a computer

Description automatically generated

Figure : Resize Virtual Machine Options

AcCessing Twitter Data

For the Virtual Machine to extract the data to analyze, it needs to legally and ethically access Twitter’s Large Databases. Luckily, Twitter encourages users to extract large amounts of information this way.

To enable this type of access to the Twitter platform, Twitter has created the “Twitter Developer Platform”, accessible through [www.developer.twitter.com](file:///C:\Users\sgupta\Downloads\www.developer.twitter.com). It is a web resource dedicated to making it easy to understand how to work with code that works with Twitter.

Under this header, a developer can find documentation about the different APIs to access and interact with Twitter. An API or Application Programming Interface is a set of functions or code that allows a developer to access applications or data.

Twitter, through their developer platform, provides access to multiple APIs that can be used to Analyze tweets, historically and in real-time, as well as actively post Tweets or send direct messages. Additionally, there are APIs available for integrating with Twitter Ads, to maximize Ad Revenue.

For accessing tweet text there are three available APIs:  Standard, Premium and Enterprise. Their difference is access level, and pricing, with Standard being the free service.

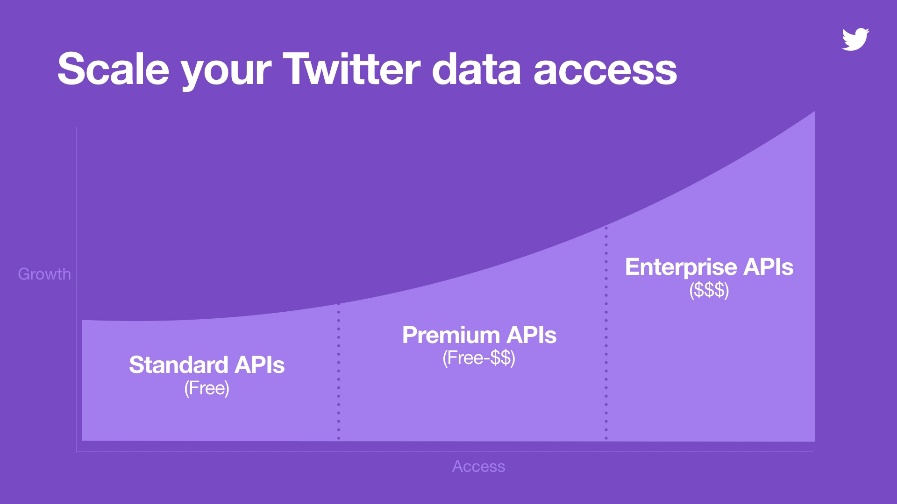


Figure : Twitter API Access chart

Twitter recommends starting out with the Standard API, and then looking at the other services when you want to scale out. The Standard API is adequate for the initial solution development, and in this way the team keeps the project’s cost low.

In order to gain Access to the organization’s Twitter *friends* data the BI team needs to complete three preliminary steps:

1. Create a Twitter developer account that’s associated with the organizations Twitter handle.
2. Register the code, application, or script that will be using the API.
3. Once the application is registered, the developer account can generate Consumer API Keys, Access Token, and Access Token Secret. These credentials are used by Twitter’s Authentication process Oauth. The developer account can change and revoke credentials when it makes sense.

A screenshot of a cell phone

Description automatically generated

Figure : Register Application

A screenshot of a social media post

Description automatically generated

Figure : Generate Authorization Credentials(Credentials have been revoked)

For fetching data in the team’s Extract and Transform and Load process, the process is utilizing Application-Only Authentication. Application-Only Authentication is a form of authentication where an application makes API requests on its own behalf, without the user context. This method is for developers that only need read-only access to public information.

Extract, Transform, and Load

Now that the team has credentials to Access twitter, they can begin using them in an application.

One of the most common ways to fetch Twitter data is through Python. Python is a good choice in language because it is robust and powerful. The language is widely used, easy to learn, and its syntax and object-oriented framework provide developers the capability to write clear, logical, code for small to large-scale solutions.

Within Python the developer team will use the TweePy library for similar reasons. It seems to be the most common way for users to access the Twitter Standard API through code. The library is well documented and well supported with a user community.

The code begins by passing the four credential values to variables.



Then it uses *Tweepy’s OAuthHandler* function to set the tokens.



Then it can access the API through a Wrapper by creating an object called ‘api’ with the *tweepy.API* function call with the authorization as the arguments.



The Twitter API Wrapper, a Python Class, has a function *API.friends\_ids*that returns an array containing the unique ids of users being followed by the twitter handle the registered application is credentialed to access.



The app can use this array to iterate through each friend. While it iterates through each friend, it will be doing all the necessary steps of the data pull.

* The app will make an additional API call to get the user screen name from their user id.
* The code will use their screen name to create the label for the flat files that will temporarily store the data.
* Then it will create the empty file, and start a write connection to that file
* Before it finishes iterating through each friend it writes the data to the flat file it created in the previous step and closes the connection to that file. With iterating through the list, it also makes the data pull and downloads the friend’s timeline of their last 3000 tweets and writes it to a flat file through the *Cursor* object.



The *cursor* object is Tweepy’s method for making a request of paginated data. Pagination is splitting data streams into pages. of data. What this means is that the code limits the results by batches. Pagination in data requests allows the code to make multiple requests to be processed, when one request fails or times out to retrieve data.

Tweepy’s *cursor* handles all the pagination work in the function so you don’t have to process a request status, the code will instead wait and try another request again.

The code calls four fields from the API.



Table : Sample Record

Later these fields are stored and imported into the data analysis tool.

The Extract Transform Load process was executed for 37 Twitter handles, roughly 10% of the organization’s list of people it ‘Follows’. For this limited amount of data, the process ran in under 5 hours. Virtual Machine Compute resources used during the Initial Load were monitored and recorded using the Metric dropdown in Azure.

A screen shot of a computer

Description automatically generated

Figure : Virtual Machine CPU Usage

The primary reason the ETL ran for such a long time was because of the rate limit of the standard API. The application is only allowed to make a certain amount of data requests every 15 minutes. The typical scenario is batches of data would be loaded in 5 minutes, and then since the application used up all its requests for the 15-minute window, the process would have to wait for 10 minutes, and then make another paginated request. The execution messages show the process slept for 138 minutes:

A close up of a piece of paper

Description automatically generated

Figure : ETL Sleeping between Paginated Requests

Since Compute percentage never exceeded 50%, the sandbox resources were more than suitable for running the ETL overnight. Disk Read/Write was also manageable when compared with some other API involved Azure ETL processes within the organization.

DATA STORAGE

Now that the process has generated flat files written out on the virtual machine, the BI team can load and store the data into SQL Server for long term storage. SQL Server is Microsoft’s Relational Database Management System Product. The data is loaded through the Bulk Insert T-SQL Command.



DATA ANALYSIS

After moving the raw data to the permanent storage location in SQL Server, it is ready to be accessed and analyzed in R libraries.

To streamline the data storage to data analysis process the R code uses a package called RODBC which directly reads from the database, and turns the data into an in-memory R data frame. Package RODBC implements ODBC connectivity, or Open Database Connectivity. This is a standard application programming interface for accessing database management systems. In this way the process skips over a flat file intermediary and inserts the data directly into a data frame called ‘my\_tweets’.



There is a known rule that is true for most of the work data analysts do. That rule is that 80% of data analysis is spent on the process of cleaning and preparing the data. (Dasu, 2003)

For the purpose of streamlining data analysis within InterDesign, the Business Intelligence team tries to prepare the data, and make it as clean as possible, and in a structure that allows the team to easily, and effectively run specific analytic tools on the data. These tools are traditional data warehousing tools designed for structured datasets within SQL Server. Text data, however, is unstructured and there aren’t any native tools in SQL Server to do deep text analysis.

For the purpose of text analysis, the BI team will rely on tools in R. However, the team of developers will still adhere to the same data preparation standards to allow them to make repeatable processes. To make these repeatable processes the team will rely on the Tidyverse collection of packages for general data wrangling, and the TidyText package for specific text wrangling tasks. As phrased by its main author, Hadley Wickam, “Tidyverse is an Opinionated Collection of R Packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures.” Hadley Wickam is the Chief Scientist at RStudio, the most widely used Development Environment of R. As he says, “The principles of tidy data provide a standard way to organize data values within a dataset. A standard makes initial data cleaning easier because you don’t need to start from scratch and reinvent the wheel every time” (Wickham, 2016)

Tidy data follows 3 rules, breaking the rules will make the data messy, or not tidy.

1. Each Variable Forms A Column
2. Each Observation forms a Row
3. Each type of observational unit forms a table

This is similar to certain traditional data warehouse principles, that the Business Intelligence team is familiar with, like Codd’s 3rd normal form, but with the “constraints framed in statistical language, and the focus put on a single dataset rather than the many connected datasets common in relational databases.” (Wickham, 2016)

For text wrangling tasks the team will rely on the TidyText package, maintained by Julia Silge. Furthermore, the developer team will be implementing many of the analysis processes introduced with the package. Like the Tidyverse collection, the code is easy to understand, and adapt to different input data structures.

When one is analyzing text, the concept of an observation and variables are harder to conceptualize. So instead, of obeying the rule “One Observation per Row” the code will obey the rule “One token per row.” And as the analysis continues, the variables will be measures that describe these *tokens*. A token is a meaningful unit of text, such as a word, phrase, sentence or paragraph, that one is interested in using for analysis,” and tokenization is the process of separating text strings into tokens.

To start, the code will “tokenize” the data into single words.

  
**%>%** is the pipline notation. DplyR, a Tidyverse package, imports this operator from another package. It is used to easily chain data analysis steps one after the other, where the output from one, is the input to another. In this case, the tweets table data structure where the data is loaded is automatically used as the first argument, *tbl*.

The first explicit argument, *word*, is the output token, the second argument, *input*, is the input data structure, and the *token* argument denotes the original token unit of the input data structure. The TidyText library, through the ‘tokenizer’ package has a specific method to tokenize tweets.



Table : Input R Data Structure

Now the data structure, ‘tidy\_words,’ contains each token, or word, for each tweet in its own line, while still preserving the original in each row.



Table : New Tidy\_words data structure

This one-token-per-row structure is different to the ways text is often stored in current analyses. For instance, a popular alternative is to use a Document Term Matrix, a useful object that is a matrix of values, with the row corresponding to the document a term appears in, the column corresponding to the term itself, and the value representing number of occurrences. But tidy text is also easy to convert to a document term matrix.

  
 The code is merely counting the combinations of words and tweets, and using a built-in function, *cast\_dtm*, to automatically create the term document matrix.

The BI Team works with these tools because of how easy they are to work with both visualization and modeling functions. Tidy Visualization tools only need to be input-tidy as their output is visual. Domain specific languages work particularly well for the visualization of tidy datasets because they can describe a visualization as a mapping between a variable and aesthetic properties of the graph. This is the idea behind Grammar of Graphics. (Wickham, 2016)

One of the easiest visualizations to show, is the frequency of tweets over time.

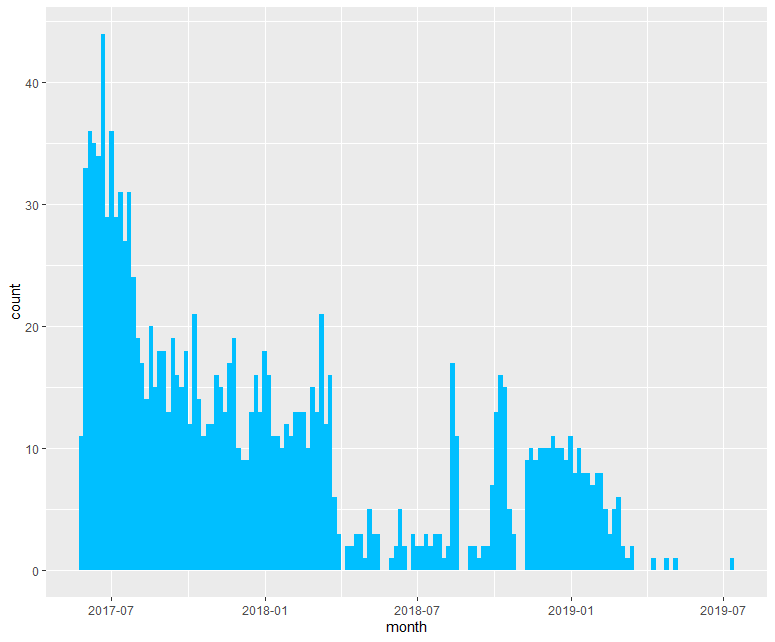


Figure : Recent tweets over time



Here the code is using the Grammar of Graphics plotting library with the histogram *geom*. *Geoms* are visual marks that represent data points, and a coordinate system. To visualize the data, the code needs to map variables to aesthetic properties of the *geom*.

In this case the code is mapping the timestamp field to the x-axis, and the count, or frequency, of the occurrences of tweets to the y-axis. With the histogram *geom* it can specify the number of bins. Bins are groups of observations. In the case with the histogram *geom* you are grouping the discrete values by their x-axis to produce the corresponding histograms. The number of bins is effectively the number of histograms seen on the visual. Since the data visualizes observations over time, bins will start and finish over equivalent time intervals.

Tidy tools can be utilized very well here because the output of one tool can be used as the input to another. This allows you to simply and easily compose multiple tools to solve different problem.

Utilizing these libraries the BI team can use the same lines of code to visualize the original dataset, and the frequency of tweets over time, or, by changing the preceding input lines, the code can look at the tokenized dataset and look at frequency of words over time. This might be useful if the organization wants to look at the mentions of a specific term over time.



The team can write code that will generate a histogram showing the frequency of mentions of a term, like ‘diy’, short for “Do It Yourself”, over time.

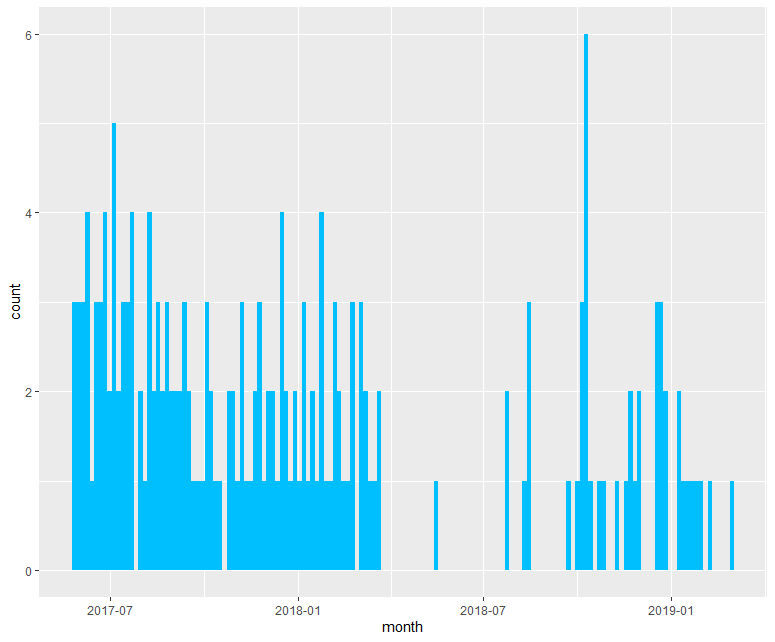


Figure : Mentions of the word ‘DIY’ over time

In this way the BI developer can utilize the pipeline to do one data wrangling step, after another, but also utilize it to do a data wrangling step before a repeatable data visualization step, or data modeling step.

Another visualization related to word frequency are wordclouds. Wordclouds are visualizations that display words based on their occurrence in a document. Words that occur frequently appear larger and towards the center. Wordclouds could also be displayed so that words that appear together are close together in the word cloud.

Creating a wordcloud is straightforward using the tidy dataset of words. The code utilizes the *wordcloud* function in the Wordcloud library. The developer can set optional parameters, on how many total words to display, with less frequently occurring words being dropped off the visualization. And the developer can choose whether or not they want the positioning to be random, using the *random.order* argument. With it being false, the most popular words will be placed in the cloud first, occupying the center, with words of decreasing frequency of occurrence being added sequentially and placed outside of the existing words area.  




Figure : Wordcloud

The R process can do sentiment analysis on the data, by joining the clean text with sentiment lexicons that exist within the package Tidytext, using it’s built in *get\_sentiments* function.



A sentiment lexicon is a database of lexical units of a language, with their sentiment orientation. In this case the Tidytext package has three sentiment lexicons, Afinn, NRC, and ‘Bing et al’, that all have single words as the lexical unit of language, just like the tokenized data.

For the Afinn lexicon, the sentiment orientation is a quantitative value indicating the measurement of how positive or negative, for the NRC and ‘Bing et al’ lexicons it is the designation of positive or negative.

By joining the tokenized twitter data to these lexicons, the code is looking up the positive or negative orientation of each word in the data. For the original tweet that might have had the sentence “I am happy,” the resulting table is one with an additional column with the sentiment value.



Table : Sentiment Join Result Data Structure

The sentiment value for the word ‘happy’ is 3. The word‘I’, and ‘am’, do not have an associated sentiment value, so they are left out of the dataset, through the inner join.

Using these libraries the code can show the sentiment orientation of a twitter timeline over past tweets.

In the code, the mod operator, **%%**, is used to group together increments of 100 tweets, by their row number. The code calls this variable, index, and determine the net sentiment orientation for each *index* as determined by the lexicons data. For the AFinn dataset it just needs to sum the quantitative value over the group.



For the other datasets, NRC and “Bing et al", this is done through the *count* and *spread* functions. Effectively it is counting the total occurrences of words for both groups, positive and negative. Since the Loughran dataset has other sentiments designations as well, like uncertainty, the code needs to filter that for only the negative and positive words. Then it splits those values into two different columns, so it can subtract them.



The code combines the results for each sentiment library into one final dataset to be plotted.



The final data product is the net sentiment for every index displayed by the three different methods.A picture containing screenshot

Description automatically generated

Figure : Sentiment of recent tweets, grouped by increments of 100

The differences between the libraries is because for certain words there are different sentiment values across the three lexicon datasets.

The R process can also produce plots to show the terms that most contribute to a positive or negative sentiment value. A picture containing screenshot, text

Description automatically generated

Figure : Words that contribute most to sentiment polarity

To generate this plot, the code looks up the sentiment values from the Afinn lexicon and sums this value over how many times the word is mentioned. To order the data in the plot, the code needs to take an absolute value of this sum, since for negative words that sentiment quantitative value is negative.

  
  
 Now that the code has imported the sentiment datasets it also can easily produce sentiment wordclouds that separate the data into the wordcloud based on the sentiment. 

Figure : Sentiment Comparison Cloud

This is done by pivoting out the data using the *acast* function, which makes it a suitable format for the *comparison.cloud* function.



The *comparison.cloud* function can be used to separate words in a word cloud based on any other variable in the data structure.

The advantages of working with text in this tidy fashion become especially clear when doing more complex analysis like one that involves quantitative modeling or regression analysis. However, modelling, even something abstract like text becomes easy. As the author of these packages writes “modelling is the driving inspiration of this work because most modelling tools work best with tidy datasets

With these tools a developer can easily do an analysis looking at the biggest change in mentioned terms. This will require the code to create a linear model for every word mentioned in any of the tweets. The linear model has time as the x-axis, or independent variable, and frequency of occurrences as the y-axis, or dependent variable.

First the code will convert the time field into the month the tweet was mentioned. Then it counts how many times each word was mentioned in that month, the total amount of words mentioned in the month, ‘time\_total’, and the total time each word was mentioned over all time ‘word\_total’. The code then filters words that were only mentioned more than 30 times over their timelines.   


Our resulting dataset, ‘words\_over\_time’, is these three amounts for every word month combination.



Table : Word\_Over\_Time data structure

At this point, the code uses the *tibble* data structure and *nest* function from the TidyR package, to nest the time series chart that exists for every word.



The resulting data structure has two columns, one for the word, and the other for the time series chart, nested within one column of the data structure.



Table : Nested Time Series Data Structure

This data structure can be printed to console.

Figure : Printed Nested Time Series

This timeseries chart can be accessed as one would normally access an item in a vector of a R dataframe.





Figure : Printed Vector Item

The code can use this nested timeseries chart to absract a linear model for each word, through the *map* function from the library purrr. The *map* function is used to apply a function to a list of items.



The function that *map* will be applying is the generalized linear model function, *glm*. The independent variable for this linear model is the ‘time\_month’ variable with the dependent variables being the word frequency usage.

Now the resulting data structure has an additional column for the nested models.



Table : Nested Models Data Structure

The *tidy* function from the broom package, along with the *unnest* function from the package Tidyverse, is used to extract the slope, or coefficients, and intercepts of the different models, into one dataframe, that it will call ‘unnested\_models’. It filters out the intercept, by only grabbing the coefficient associated with the ‘time\_month’ term.



The ‘unnested\_models’ data set now has no nested data, and instead has model slope, coefficient, intercept, statistical and p-values, in additional columns.



Table : Unnested Models Data Structure

Then the code can filter out only significance slopes by using the same filter function it used earlier.   


Now to plot the data the code just joins this model dataset back to the 1st dataset, with the timeseries charts, to display the timeseries data for only words with significant statistical models.  
A close up of a map

Description automatically generated

Figure : Terms that have had the most change in mention rate over time

Another analysis the organization can do is to look at the n-grams that exist in tweets. N-grams are contiguous sequences of ‘n’ items from a given sample or speech. So far, the process has been working with words, which are 1-gram, or unigrams. A 2-gram, or bigram, is a two-word phrase. For instance, the bigrams that exist in the sentence “Hi how are you” are “Hi how,” “how are,” and “are you.” Changing the original ‘unnest\_token’ output token parameter to ‘ngrams’, the developer can select the specific n-gram that they want to see, by the parameter *n*. With this code the developer/analyst can see the most common trigrams, or three-word phrases, that exist in different tweets.



A close up of a logo

Description automatically generated

Figure : Most mentioned 3-word phrases

Using the frequency of occurrences of bigrams, or two-word phrases, the code can be integrated with network packages to visualize the network of connected terms. First the code needs to split the bigram into its two component words.

Now the code creates a graph, which is basically a set of vertices and edges that connect those vertices. In this case our vertices are words, and the edges denote they exist in a bigram together.

Utilizing the *graph\_from\_dataframe* function from the iGraph package the code can create our graph data structure.



This can be plotted into a network using the GGraph package which extends the Grammar of Graphics library to network and graph data. A close up of a map

Description automatically generated

Figure : Bigram Network

REPORTING

To publish a standard report that people in the enterprise can open, without additional licensing, the BI team has utilized R markdown to “knit” an html file, and a Word Document. R Markdown allows a developer to publish R output, code, and annotations from code embedded in a file, of extension ‘.rmd’. In this case the output are the visualizations shown in the analysis section.

Sample reports of Twitter handles can be accessed here: [rpubs.com/SalilUWDS](http://rpubs.com/SalilUWDS).

REPORT USAGE

Now with the processes created, an individual in the company could access the report and answer the following questions:

* Are influencers still frequently posting?
* Are influencers posting positive content?
* What is their content about?
* What type of language, and high impact words are they using?
* Is there a trend in the language they are using, or the ideas they are mentioning?

In this way the business is many steps closer to evaluating growth in social currency associated with their brand through text analysis.

Conclusion of Development

At the conclusion of the initial development period the Business Intelligence team has designed processes to access, load, store, analyze and report Twitter timeline data. The code and processes created are easy to understand and documented. They are fully contained in a sandbox environment provisioned through Microsoft Azure. This sandbox environment can be expanded or migrated through the flexibility of cloud technology.

There are inherently three major software pieces: Data access, extract, and load in Python, data storage in SQL Server, and the data analysis and reporting using R. The processes are versatile to handle more custom requirements around the visualizations and amount of data processed. Additionally, most of the concepts applied, and the processes developed can be applied to other useful text data sources within the organization.

NEXT STEPS

1. Building out more analytical function: With ongoing development it would be useful to build out more analytic functionality to the process. There are more powerful analytical methods for text analysis, or natural language processing. Examples of these include Name Entity Recognition or Topic Modeling.
2. Load more Twitter data: An obvious next step is to fix the load process to access more data. Developing a load solution using the Premium API would be necessary.
3. Actionable analysis: If the BI team can make the report more actionable it would benefit the business. A report that is actionable is one that when the report user glances at it, they quickly know what action plan to take to benefit the business.
4. Static reporting to dynamic reporting: The reporting is static images, it would be beneficial if the BI team can begin exploring ways for users to interact with the data, for instance, filtering out values based on user or content.
5. Integrating more text: Much of this process can be used with other text sources that might show strong indication of positive or negative feedback.
6. More seamless automation: The process can be improved with more seamless integration and automation by using new product capabilities of SQL Server and Azure.

* Utilizing Python script task in SSIS: SSIS, Microsoft’s ETL tool for SQL Server, has the capability of running a Python script, so this would enable the team to start the Python ETL through SQL Server Jobs and Integration Services Packages.
* Utilizing R Services on SQL Server: SQL Server has the capability of running R code through T-SQL stored procedures. Utilizing this would allow the team to kick off the R analysis code through SQL Server Jobs.
* Utilizing SSIS for accessing the Twitter database: SSIS has the ability to make OAuth REST API calls directly, so using this could potentially bypass Python code altogether.
* Adaptation of a more Platform-as-a-service model would mean the organization would bypass virtual infrastructure, and instead use a virtual service. There are multiple virtual Azure services the organization could integrate giving the BI Developer team the capability of orchestrating data movement through the Azure Portal, as opposed to logging into a Virtual Machine.
  1. Azure Storage Service
  2. Azure App Service
  3. Cloud SQL database Service
  4. Azure Power BI Reporting

REFERENCES

Apostu, A., Puican, F.C., Ularu, G., Suciu, G., & Todoran, G. (2013). Study on advantages and disadvantages of Cloud Computing – the advantages of Telemetry Applications in the Cloud.

Eising, P. (2017, December 07). What exactly IS an API? - Perry Eising. Retrieved from <https://medium.com/@perrysetgo/what-exactly-is-an-api-69f36968a41f>

Dasu T, Johnson T (2003). Exploratory Data Mining and Data Cleaning. Wiley-IEEE.

Dholakiya, P. (2017, April 28). 3 Things a Brand Can Do to Become a Household Name. Retrieved from https://digitalbrandinginstitute.com/become-household-name/

Grolemund, G. (2016). R for data science. Beijing; Köln: OReilly.

Klein, R. (2018, May 15). The Value of Social Currency - Marketing Today. Retrieved from https://medium.com/marketing-today/the-value-of-social-currency-d064629f8140

Oza, H. (2018, July 25). 10 Advantages Of Digital Marketing Over Traditional Marketing. Retrieved from https://prowly.com/magazine/advantages-of-digital-marketing-over-traditional-marketing/

Roesslein, J. (n.d.). Tweepy Documentation. Retrieved July 15, 2019, from https://tweepy.readthedocs.io/en/latest/index.html

Silge, J., & Robinson, D. (2017). Text mining with R: A tidy approach. Sebastopol, CA: OReilly Media.

Silge, J., & Robinson, D. (2019, May 16). Text Mining with R. Retrieved from https://www.tidytextmining.com/

Wickham, H., & Grolemund, G. (n.d.). Tidyverse. Retrieved July 15, 2019, from https://www.tidyverse.org/

Wickham, H. (2014). Tidy Data. Journal of Statistical Software, Retrieved July 15, 2019, fromdoi:<http://dx.doi.org/10.18637/jss.v059.i10>

Wickham, H. (2016). R for Data Science: Visualize, Model, Transform, Tidy, and Import Data. OReilly Media.