Universidade de Brasília

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Monitorando e entendendo a eleição brasileira por meio de Processamento de Linguagem Natural

Teógenes Moura

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia da Computação

Orientador
Prof. Dr. Vinicius Pereira Gonçalves

Brasília
2019

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Monitorando e entendendo a eleição brasileira por meio de Processamento de Linguagem Natural

Teógenes Moura

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia da Computação

Prof. Dr. Vinicius Pereira Gonçalves (Orientador)
ENE/UnB

Prof.a Dr.a Marisa Von Bulow
IPOL/UnB

Prof. Dr. Geraldo P. R. Filho
CIC/UnB

Prof. Dr. José Edil Guimarães de Medeiros
Coordenador do Curso de Engenharia da Computação

Brasília, 4 de Julho de 2019

Dedicatória

Dedico essa dissertação aos meus pais, cujo apoio é fundamental, a minha irmã, que mesmo com todos os obstáculos de saúde sempre se dispõe a colaborar em todas as ideias, mirabolantes ou não, do irmão, aos amigos, cuja presença é de extrema importância mesmo em momentos que parecem sutis mas são indispensáveis para o prosseguimento dos esforços, e a Cookie, meu gato, fiel escudeiro e companheiro nas múltiplas horas dedicadas a esse trabalho.

Agradecimentos

Primeiramente, agradeço a todos que apoiaram essa jornada de algum modo. Desde aqueles cuja conversa não durou mais que alguns minutos mas muito produtivas, até aqueles que ofereceram seu tempo e muita paciência com as muitas dúvidas de prosseguimento no curso deste que vos escreve. Em especial, papai e mamãe, que tentam ajudar como podem mesmo sem entender bem o por quê do filho graduando em Engenharia de Computação falar tanto de ciências sociais, Sara, que não falha em garantir que o irmão relaxe em momentos ocasionalmente muito tensos, aos amigos mais próximos que viram os muitos altos e baixos dos anos gastos na UnB.

Além desses, agradeço também a Darcy Ribeiro por estabelecer na Universidade de Brasília, dentro dos seus limites e possibilidades, uma das poucas universidades públicas brasileiras que dão boas vindas a aqueles que não conseguem se aquietar nos limites de suas origens acadêmicas. Sem o fluxo constante de ideias, conversas e risadas entre os departamentos de Ciência da Computação, Engenharia Elétrica e Ciência Política, esse trabalho jamais teria sido realizado.

A professora Marisa Von Bulow, e a todos os integrantes do Resocie, por acolherem em seu grupo um visitante exótico provindo das terras das ciências exatas. Sem essa orientação, que perpassa os limites acadêmicos, os anos de universidade teriam sido fundamentalmente menos interessantes.

A meu orientador Vínicius Pereira Gonçalves, que aceitou trabalhar num projeto a priori alheio a sua realidade e que apareceu de modo certamente repentino.

A Alan Turing, que é a prova máxima produzida pela humanidade de que são as decisões que tomamos que mostram quem somos, e que esforço e talento independem de cor, raça, orientação sexual, e tantas outras características inatas que ainda separam pessoas nos dias atuais.

Resumo

A discussão acerca dos efeitos das tecnologias digitais sobre a democracia passaram a receber muita atenção desdde o advento das mídias sociais e dispositivos móveis. Neste documento, nós entramos na discussão sobre mecanismos de busca e seus efeitos políticos, cuja importância aumentou significativamente após a eleição norte-americana de 2016. Nosso estudo foca na eleição geral brasileira de 2018,um processo bastante conturbado que levou a eleição de um Presidente de extrema direita. O artigo primeiro descreve o processo de aquisição dos dados. Nós construímos um processo de treinamento-busca-coleta no qual criamos contas avatares com a intenção de representar, o mais fielmente possível, eleitores dos espectros políticos da esquerda e da direita. Então, utilizamos um processo automatizado para fazer buscas de modo que o Google pudesse entender as diferenças entre perfis. Por fim, repetidamente coletamos os resultados mostrados a cada usuário durante o período da eleição, baseados numa lista geral de termos de buscas, que resultou num banco de dados contendo aproximadamente 300 mil URLs. Utilizamos o algoritmo Word2Vec, uma técnica que nos permite observar quais palavras e frases estão mais proximamente associados a tópicos sensíveis da eleição, como 'Fernando Haddad' e 'Jair Bolsonaro'. Coletamos uma base de 2 milhões de palavras e conseguimos demonstrar o uso de retórica violenta em ambos os lados da disucssão, com palavras como 'medo' e 'agressão' aparecendo relacionadas a Fernando Haddad, enquanto 'inimigo' e 'nazista' aparecem próximas a Jair Bolsonaro, o que demonstra uma eleição extremamente polarizada.

Palavras-chave: Word2Vec, NLP, mecanismos de busca, democracia

Abstract

The discussion around the effects of digital technology on democracy gained the spotlight since the rise of social media and mobile devices. In this paper, we shed light into the discussion around search engines and their political effects, which gained a lot of momentum after the 2016 US Election. Our study focuses on the Brazilian General Election of 2018, a highly disruptive electoral process, which led to the election of an extreme right-wing President. This paper first describes the process of gathering the data. We set up a training-searching-collecting framework in which we created avatar accounts intending to represent, as accurately as possible, the digital behavior of voters belonging to the right and left spectrums of the political debate. Then, we used an automated to approach to make queries on their behalf so that Google understands the differences between the profiles. Lastly, we repeatedly collected the results shown by Google to each user during the election period, based on a common list of search terms which result in 300 thousand URL records in our database. We then analyzed the titles of the URLs shown by Google, as well as the contents of the texts of each link in the results. We used the Word2Vec algorithm, a Natural Language Processing technique which allows us to determine words and phrases closely associated with key topics in the election, such as the main Presidential candidates' names: Fernando Haddad and Jair Bolsonaro. We collect a dataset of more than 2M words and are able to demonstrate the use of violent rhetoric on both sides of the discussion, with words such as 'fear' and 'agression' appearing closely related to Fernando Haddad, while 'enemy' and 'nazist' are seen next to Jair Bolsonaro, which are results that clearly demonstrate an extremely polarized election process.

Keywords: Word2Vec, NLP, search engines, democracy

Sumário

Lista de Figuras

Capítulo 1

Introduction

1.1 Context

The objective of this research endeavor is to observe if search engines' results vary depending on the political leaning of a given user during the period of an election.

Computer-based technology is now one of the driving forces of economies around the globe and a pervasive facet of society in the 21st century. Cloud computing and Moore's law[1] have allowed companies and developers to design and build products that scale massively in a fraction of the time companies in other industries take to conquer global markets. Examples of such scenario are manifold: Companies such as Google, Facebook, Amazon and many others deploy products for hundreds of thousands of users worldwide in real time, often leading to eye-boggling revenue. The discernment between technology and magic therefore becomes evermore difficult to find, as smartphones become as ubiquitous as people themselves and AI-driven applications become more and more human like.

In the midst of the huge technical development society has seen in the last 20 years (we need only remember that companies that are now perceived as everlasting were founded less than 30 years ago, such as Google (1995), Facebook (2004) and Twitter (2006)), the political landscape has also seen a multitude of drastic changes all over the globe. Large democracies have faced challenges which were heretofore novel-exclusives such as protests being organized through social media, AI-powered mass surveillance and the deeply important discussion about technology and its place in our society, taking into account political discussion this day and age happen primarily as streams of 1s and 0s flowing through intricate webs of computers spread across all continents. As such, computer-based technologies have taken the spotlight as a fundamental asset for democracies to endure as

[1] Moore's law states that the amount of transistors per area unit in a computer processor doubles every 18 months, and was proposed by Gordon E. Moore, who was CEO of Intel by the time he stated the claim. More on https://www.cs.utexas.edu/ fussell/courses/cs352h/papers/moore.pdf, access on February 28th, 2019

well as to fall. To discuss technology without considering the users as well as the engineers who built the fabric of the digital universe leads only to a frail, superficial meaningless monologue.

Such important role notwithstanding, technology has yet another impacts which are even harder to measure and discuss. To acknowledge a system was breached, even though hard to do requires a clear answer. Either it was or it wasn't. The same as the inner workings of the electronic devices we now share our most delicate intimacies with - either a cellphone works or it doesn't. Such binary mindset cannot be applied to the end users of such devices. Research is still uncertain on the impacts of our constant electronic dependency in our brains and in the cognitive development of children. It has been indicated to the international community that children absorb more microwave radiation than adults[1] and that spending long periods of time on media devices may bring harm to children and teenagers[2]. However important health concerns are, there's one aspect in which the scientific community is very much still in the dark: How will technology shape the future and the present of power relations and political disputes? How can we guarantee that information wars aren't being fought in the background of political campaigns aiming to deliver information about one candidate to voters and stopping other candidates to reach the same audience? Furthermore, do people really have access to trustworthy news sources which have truth and honor as their core values or misinformation is already so widespread that the way-back journey is no longer an available path?

It's clear such questions are fundamental to our society and may help shape discussions for generations to come. It's reasonable to argue that any person claiming to have objective answers for such subjective questions is either exceedingly naive or ill-intentioned. This paper contributes to the overall debate about the political impacts of Technologies. Because this is of course a very broad debate, it focuses on the role of search engines and the provision of political information. .

The relevance of this research effort derives from the increasingly important role search engines have played in the provision of political information. In spite of this relevance, we still know little about how to study such roles and their impacts. To have almost infinite information is not enough: For any person to make good use of it, it needs to be indexed and organized in a way a human brain can make sense of. In thinking the internet, the following quote comes to mind:

We are all now connected by the Internet, like neurons in a giant brain.

which was made famous by Stephen Hawking. It reflects an interesting side to the human nature already observed in other fields of knowledge: The tendency to observe nature in order to replicate its intricacy and complexity, taming it to our own well-being. Examples of this might be when mechanical engineers design machines based on animals found in the nature, musicians apply Fibonacci sequences to their compositions, photographers rely on human eye-inspired cameras, and poets hone their craft around the structure of spoken word, shaped by years of evolution, in order to make reality slightly better than it really is even if for brief moments. One less explicit appearance of such a phenomenon happens in systems which are hard to observe by the naked eye. Electrical engineering professors have complained for years on the difficulty of teaching a subject which students aren't able to see the inner workings, only wonder. The same happens with the internet: As we use our cellphones, or watch movies on a computer, or call a cab, or order a pizza online, it's only natural to forget the huge amount of complex operations that take place in order for a single button to work.

That is, therefore, to say that every single person connected to a network of computers - specially the largest one of them - acts similarly to a neuron in the human brain. As much information a person can process or create, all he or she can do is to send to and receive it from other neurons to which they're connected through network routers, ocean cables, satellites, data centers and so on. It, in a way, makes it easy to privilege one's own individuality: No user follows the trail of information generated when a request made to a web page happens[2] to see how everything comes to be. What usually happens is they'll lock their smartphones and hope it works the next time they try unlocking it. This can lead to a sensation that everything is a closed experience which doesn't necessarily bestows a huge influence on the society as a whole or even to the person sitting next to them. Food-delivery and cab services might offer the experience of *"breaking the fourth wall"*, but their context is extremely limited in scope compared to the effect web technologies might cast on deeper social structures.

Discussions on the impact of algorithms in our daily lives are ongoing and findings are fascinating to anyone interested in the intersection between Computer Science and society.Developers who design and build applications we use in our daily routine are just as important as the end users in understanding the role of technology in a social setting. Such importance derives from the fact that to program is to express one's view of the world in terms of mathematical models which are translated to code and deployed as final products for people to consume. Therefore, every developer is unique in his or her ways to write code, despite standards responsible for keeping cohesive code bases. It doesn't pertain only to the act of writing code itself, however: datasets we choose to train machine learning models on (which have become exponentially important over the years and are now the stepping stone of AI-based applications) also carry implicit biases not necessarily clear to those assembling them or engineering programs with them.

A clear example of such scenario happens when we develop applications that deals

[2] At least not the ones graduating in Computer Engineering programs

with racial issues either explicitly or not. A team at the MIT Media Lab has demonstrated machine learning algorithms to reproduce racial biases of developers. A commercial facial-recognition software introduced an alarming disparity in it's capability to correctly recognize faces of people of white and black heritages. The error rate for white males was 0.8% in it's maximum, while women with dark skin tones were misclassified up to 34.7% of times[3]. In a society in which discussions around violence perpetrated by police officers have led to mass protests and deaths, it is shocking technology might aid the perpetuation of such scenario because of racial bias coming from the developers.

Another case of technological interference in social processes attracted the attention of international media organizations during the 2016 US election of president Donald Trump. Russia's interference in the electoral process is still being discussed, but numerous findings were reported by various news sources since the election. The New York Times reported the CIA having evidence of Russian effort to affect the elections earlier than the presidential campaign[4] and later produced an extraordinarily detailed summary of all actions believed to have been taken by Russian authorities in order to help Donald Trump's campaign, which ranged from using bots on social media websites downright to raw computer hacking[5].

The aforementioned cases compose a tiny fraction of all instances where technology carried a key role in enabling actions directly related to the structures of political power and government around the globe, and we could go on tirelessly. Attractive idea as it seems, however, it's of no help to the scientific community if we attain ourselves to the telling of past incidents. It's much more desirable to explore and try to explain or prevent future wrongdoings caused by technology or to unravel paths which will lead to stronger democracies and popular participation.

1.2 Goals

In this document, we're entering a discussion which notwithstanding it's complex technology is also very controversial from the social sciences perspective. The reason for such compound difficulty stems from the fact both point of views are extremely recent and most research groups have only taken up to projects like this one for only a few years at most. From a technical point of view, search engines are extremely hard to build. It's taken Google - which owns the highest market share - thousands of engineering hours to come up with a product which is trustworthy and efficient result-wise. Competitors have spent similar resources to catch even tiny fractions of the search engine market. Providing top-notch results depend not only on algorithmic expertise, but also on storing and utilizing tremendous amounts of data the best way possible. As one can expect both parts are proprietary in the vast majority of cases, making it extremely hard for outside researchers to truly know how search mechanisms work. Also, Google, for example, is known to make modifications to it's search engine constantly thus making it unlikely for any one person to know exactly how it works at any one point in time.

At the same time, social scientists have been working hard to understand how web technologies affect human behavior. Clearly, different fields of knowledge will explore their own problems of interest but their shared difficulty is more or less the same: We're living a period of unforeseen impactful societal changes in an extremely rapid pace, leaving researchers as if trying to drink water from a fire-hose. One should only remember current technology powerhouses weren't even born some 15 - 20 years ago. Uber, the ride-sharing giant, was founded in 2009 and targets a valuation of $120B for 2019, only a decade later[3]. Tinder, the dating app, was born in 2012 and has matched people over 20 billion times since its release[4]. We could continue this list almost indefinitely, but the main point is that while it seems to be evident social-structure changes coming from those companies, it remains unclear how such impact looks like, what it affects and why it does so.

In the next sections, we won't assign us the task of answering all these questions. Instead, we're going to focus on one aspect of a possible impact deriving from the use of search engines during election periods. We'll try to take a grasp into understanding how people inform themselves politically during campaign and election periods and if search engines - with a special focus on Google - contribute to the information or disinformation of users from varying political leanings. One observation is of uttermost importance however: we're **not** entering the bias discussion even if we mention it throughout the text. Such a decision comes from an understanding inherent to analyzing impacts of proprietary technology. It'd be extremely hard for us to determine causality relations without access to what might be the reason for a cause/effect relationship behind the curtains: the actual algorithms powering the search engine. As such, we'll adhere ourselves to the perspective of debating information and disinformation, delving our feet into technical waters when the matter calls upon such discussion and guiding ourselves from the perspectives of both Computer Science approaches (including analogous areas such as Artificial Intelligence and Natural Language Processing) and the Social Sciences.

[3] https://www.theguardian.com/technology/2018/oct/16/uber-targets-120bn-valuation-2019-flotation-report

[4] http://www.businessofapps.com/data/tinder-statistics/

Capítulo 2

Theoretical discussion

To discuss a problem which concerns at least two vastly different fields of human knowledge such as technology-based fields and the social sciences require strong theoretical foundations from both perspectives. In this sessions, we'll discuss concepts which will be of importance to the remainder of the document, where we'll analyze data sets under the light of social sciences in order to understand how technology is helping to shape our relation as a society to structures of power.

2.1 The role of information in 21st century western societies

2.1.1 Influence of Search Engines on modern democracies

Sorting numbers is a problem which might seem trivial to the average person, but one exciting for Computer Science enthusiasts. The challenge of finding the quickest, more efficient method of keeping information in order has led to multiple approaches, each with their own strengths and weaknesses. As an example, the Bubble sort algorithm is widely regarded as an inefficient algorithm: It's time complexity is $\mathcal{O}(n^2)$ for the worst case[1]. It means that it will use quadratic time to sort an array of numbers in the case the original array hasn't got any previous sorting. Although the previous sentence might not make a lot of sense to the casual reader, the intersection between Computer Science and Social studies have become so hard to disentangle that even the former president of the United States, Barack Obama, has his own opinion on the Bubble sort algorithm. In an interview with Google's CEO, when asked which was the best way to sort 1 million integers, he said:

[1] An interesting paper on the history of Bubble sort can be found here: https://users.cs.duke.edu/ ola/-papers/bubble.pdf

"I think the bubble sort would be the wrong way to go".

[2]

While we're at the topic of Obama and Google, search has been a problem as fundamental to computer scientists as sorting. Over the years, much like sorting, it has seen multiple approaches with various levels of efficiency. Linear search, Binary search, Jump search are just a few examples of what could be a rather long list. There is, however, one aspect of the search problem which makes it weirdly strange: If you ask the average citizen, he'll probably name effortlessly the largest search engine ever built: Google.

Google's mission, according to it's founders, is to "organize the world's information".[3] It was launched by two Stanford PhD students as an academic project, which introduced the algorithm developed by them and which would later change the whole search industry: Pagerank. Previous search mechanisms, such as Altavista and Yahoo, used to implement less complex search algorithms, which wouldn't make a lot of effort in order to understand the context in which each webpage existed, as they'd just create a ranking of webpages by simple criteria and ask users to perform complex queries in order to find what they were looking for.[4]

Pagerank, however, works differently: It analyzes connections of webpages to each other as well as their content, assigning a score to each of them.It, in turn, dictates the placement the page will get when a user typed a relevant query in the search box. The original paper written by Sergey Brin and Lawrence Page, Google's founders, outlines Pagerank as follows:

"We assume page A has pages T1...Tn which point to it (i.e., are citations). The parameter d is a damping factor which can be set between 0 and 1. We usually set d to 0.85. There are more details about d in the next section. Also C(A) is defined as the number of links going out of page A. The PageRank of a page A is given as follows: PR(A) = (1-d) + d (PR(T1)/C(T1) + ... + PR(Tn)/C(Tn)) Note that the PageRanks form a probability distribution over web pages, so the sum of all web pages' PageRanks will be one[6]

The underlying math goes far beyond the probability formula in the excerpt, and involves linear algebra, matrices and vectors, which we'll not dive into here. The important characteristic, however, is clear: The underlying assumption of the Pagerank algorithm is that if a web page is important for an specific matter, it'll be mentioned by a lot of

[2] https://www.wired.com/2007/11/obama-elect-me/

[3] https://www.google.com/search/howsearchworks/mission/

[4] An interesting anecdote of how Google took Altavista's place can be found in this Quora answer: https://www.quora.com/Why-did-Altavista-search-engine-lose-ground-so-quickly-to-Google

other pages, which allows the problem to be modeled as a graph problem. With this approach, the search problem gains a multitude of mathematical tools unavailable to other contestants.

The popularity gained by Google ever since it was launched is no secret: It's market cap is currently $758.75[5], with an increase from 334.56B in January 12th, 2015 to a maximum $882.35B by July 26th, 2018. Aside from financial values, the company has been widely successful at establishing itself as one of the landmarks of the digital era: It's been ranked the best place to work at least 6 times by Fortune[6] and even inspired a Hollywood movie whose plot was a satire of the famously difficult interview process at the company[7]. All the work put into making it a "cool place to work"has produced results: Google's global brand value achieved a record $302.06B[8] in 2018, which reflects the impact it has on the collective imaginary of populations around the globe.

Such serendipity, however, needs to be put into perspective: Is it really safe to trust the search engine will always act in user's best interests and produce results which are as meaningful as they should be? Google's current CEO, Sundar Pichai, has recently announced the company would move from a mobile-first philosophy to an AI-first strategy[9], which only serves as a glowing reminder of what we've discussed so far: In an AI-driven world, how valuable is user privacy in contrast with their personal data? Is there a healthy trade off to be made in this aspect or does it always need to be a zero-sum game which favours large tech companies?

Google has faced multiple complaints due to privacy issues, including lawsuits which received extensive media attention, such as the case of Google Italy and a video uploaded to the platform which depicted a boy with autism suffering bullying from his colleagues at school. After being uploaded, it remained online for at least 2 months before it was taken down by the company, regardless of the many requests for deletion submitted by the community. In this case, should the company be considered liable for content uploaded by it's users? Or all guilt must be cast in those responsible for creating and uploading the video to the platform? Italian jury has found 4 Google's executives guilty in the case, as put in

[5] According to https://ycharts.com/companies/GOOG/market_cap, access on January 27th, 2019

[6] http://fortune.com/best-companies/2017/google/

[7] *The Internship, 2013 - https://www.imdb.com/title/tt2234155/*

[8] According to https://www.statista.com/statistics/326046/google-brand-value/, access on January 27th, 2019

[9] *"We will move from mobile first to an AI-first world", as Sundar wrote on his letter to investors, available at https://blog.google/inside-google/alphabet/this-years-founders-letter/, access on January 27th, 2019*

"all four Google executives were acquitted with regard to the charge of defamation, and three of them were sentenced to a six-months suspended jail sentence for violation of data protection law.

[7]

Such result reignited the discussion which was ongoing in Italy regarding the defamation of public personas, such as politicians online. While privacy is an enormously component of the debates regarding the search giant, several other problems involve the Mountain View company. From obscure relationships with governments[10], to data breaches [11], to extremely controversial projects, such as the one to launch a search engine in China, where censorship is supported by the government (the so called "Project Dragonfly")[12], it is clear that the company's activities are not always unanimously harmless when considering the diversity of users they serve, their business interests, their shareholders, leaders and political relationships.

2.1.2 Search engines and bias

One of the main concerns about Google's activities is how they handle biased results if present. In spite of interest for multiple groups, one quickly comes to mind when the bias discussion arises: Politicians. They're one of the most affected groups by biased search engines, since a left-biased search mechanisms is clearly a threat to the election of right-wing candidates and vice-versa. In a hearing at the congress held on December 11th, 2018, representatives of both the Republican and Democrat parties expressed their concerns on the matter.

Republican senators questioned Sundar Pichai, Google's CEO, repeatedly on the matter. As an example, we quote an excerpt from The New York Times:

"Texas Rep. Lamar Smith tried to needle Pichai with a series of studies and statistics claiming to show suppression of pro-Trump viewpoints in Google search results. Smith cited a claim from conservative outlet PJ Media that 96 percent of results for a search on news about Trump were from left-wing media and findings from psychologist Robert Epstein that Google could have swung 2.6 million votes in Hillary Clinton's favor during the 2016 election.Pichai responded that Google had investigated the specific findings, which allowed him to pivot the line of questioning to a debate over the studies' methodologies all while maintaining that Google in no way discriminates against conservatives". "[13]

[10] https://www.forbes.com/sites/davidpridham/2017/07/19/how-google-tries-to-buy-government/, access on January 27th, 2019

[11] https://www.theguardian.com/technology/2018/oct/08/google-plus-security-breach-wall-street-journal, access on January 27th, 2019

[12] https://www.bbc.com/news/technology-46604085

[13] https://slate.com/technology/2018/12/google-hearing-sundar-pichai-republicans-conservative-bias.html, access on January 27th, 2019

On a similar note, another Republican senator, Rep.Steve Chabot, questioned the executive on why he saw a prominence of negative results about him online during the campaign period, to which the CEO presented the argument that neither he or any one or a group of employees had the ability to modify search results without going through an extensive procedure devised to prevent ill-intentioned employees to affect the quality of the service provided by Google.

The list of questions about political bias made during the hearing is far from over, but both presented are capable of exemplifying the extension of the concern posed by the matter to representatives and candidates. The other side of the coin, however, is equally as important (if not more): Is the population getting biased results due to their political leanings? Is the California-based company able to provide users with clean, resourceful results which actually carry meaning and inform voters rather than support misinformation?

Both questions are still unclear at this moment and research on this matter, although recent, is already gaining traction due to the relevance of the topic and the exposure it gets in traditional media. Despite not being a search engine, it has been shown that running online messaging campaigns on Facebook do alter offline user behaviour, including tendency to attend the election itself (we just need to remember that in the US voting isn't mandatory). An experiment run with 61 million people has shown that users who received political mobilization messages on the platform were more likely to express themselves politically and to actually go cast a vote on election day. More precisely, they ague their efforts increased turnout by 60,000 votes and indirectly affected 280,000, which together represent 0.14% of all registered voters[8]. Social media and Twitter have also been studied as possible networks which could have the potential to influence decision making process, leading to varying results, such as in [9] and [10].

There has also been research following data extracted not necessarily from Google itself, but from one of it's other services, such as Google Trends, for example. [11] has discussed the relationship between Google's search volumes and real life events both in the US and UK. The paper shows that populations in both countries react differently when it comes to their behaviour online in relation to what happens offline: Americans showed a tendency to use extreme events (such as political gafes) as a trigger for more in-depth research on the opinions and policies a candidate representative has in relation to the theme relative to the gafe. If a politician makes a so called joke involving a minority group, for example, Google Trends in the US logged an increase in searches relating womens rights and the candidate, for example. UK crowds also seem to be more affected by TV debates than US's, which could lead to interesting findings on the relationship between offline events and online behavior during political campaigns[11].

2.1.3 Related work

Other researchers have concerned themselves with the role of Google Trends as a possible predictor of the result of an election, to varying results. It has been found that Google Trends isn't generally a good source for deriving results before the election actually happens, but in some very specific cases it might provide good predictions.[12]

A more general study has implemented a custom-made search engine and deployed it to groups of users in India and the US in order to understand the effect a biased search engine imposes in a decision making procedure. The findings, if replicated by following studies, are alarming: It was found that a biased search engine is able to mold the political preference of an undecided voter in 20% of the cases, and more shockingly, a considerable portion voters who were told their result were purposefully biased by the researchers reported they'd still trust the search engine.[13]

If such a profound effect is found in a controlled environment such as a custom-tailored search engine with a small amount of participants (when compared to a massive, global search engine as Google), we can only wonder what happens when a search engine that processes more than 1 trillion searches an year[14] is active and operating during all periods of an electoral process, from campaign up until the results are made public by election authorities.

In parallel to researchers concerned with the search engine spectrum of our discussion, there are also many relevant works that employ the Word2Vec algorithm as a fundamental stepping-stone for performing high-quality analysis of bodies of texts. As an example, one interesting application was to build a Sentiment Dictionary using as an underlying structure the Word2Vec algorithm, which served the purpose of understanding the emotional context of messages sent through Weibo, the largest micro-blogging company in China[14].

With this question in mind, in the next section we'll perform an exploratory analysis conducted during Brazil's 2018 general election and explore whether search results vary according to a person's political leaning during the campaign period or not.

[14] http://www.internetlivestats.com/google-search-statistics/, acess on January 27th, 2019

Capítulo 3

Methodology

3.1 Understanding text-based data

Throughout this text, we aim to contribute to the discussion around search engines and their role in society. To do so, we need to go beyond the theoretical discussion and make our best to understand our context in the most analytically way as possible so that we're able to backup our findings and insights with reliable data coming from the search engines themselves. Luckily, over the span of the past few years, high-efficiency computer resources such as Cloud computing AI processing have become ever cheaper. Such scenario led researchers to be able to process tremendous amounts of data which weren't possible to be understood only a few years prior. As a result, new findings and discoveries began to arise in various Computer Science areas, such as Artificial Intelligence and Natural Language Processing.

In the Methodology chapter, we'll use several NLP algorithms and ideas, so in this section we'll give a brief introduction to core concepts in the field so the reader is able to follow regardless of previous experiences with text processing algorithms.

3.1.1 Introduction to Natural Language Processing

To communicate with one another is of the reasons for the widespread of the human species. Language has allowed us to overcome natural predators larger and more powerful than any one human being, but still susceptible to groups of intelligent people talking to each other through a predefined set of words which carried meaning to them.

Over history, language has become more and more important to understand the inner workings of a given society, as well as their vulnerabilities, ambitions and valued behaviors. To study a society, therefore, also means to study their language and how it is structured.

It is clear languages possess a grammar through which they're structured, and also that this grammar is reflexive of what that society perceives to be important. It is also known people will bend those rules as they see appropriate to make communication easier and/or more effective, through the usage of slangs for example. Computers, however, lack the capability to easily understand nuance and subjectivity, both necessary for adapting language to a certain specific context. Computers - as we understand them at this point in time - can only grasp the meaning of numbers, and more specifically only ones and zeroes. It is needed then that a person who is set out to understand language through computational means also needs to find a way to represent a *corpus*[1] as an entity which we're able to manipulate mathematically.

The approach we're taking in this text - and the one which seems to be the prevalent in natural language processing these days - is to understand the text statistically, which means that instead of trying to hard code a set of grammar rules to a computer and try to understand language through those rules, we believe it'll be of much better use to resort to counting words. At first glance, it might seem as a naive approach. After all, what good can it be to sit around counting words? It so happens, however, that understanding language also means to understand the context through which it happens, and such context is expressed not necessarily in the rules that form a language, but rather in how people break them.

In the next sections, we'll present some techniques which are fundamental for the kind of analysis we'll perform for the duration of this text. We don't intend to go too deeply into each of those topics, as such coverage would go beyond the scope of this document and require a degree of mathematical rigor which doesn't necessarily help us to make this text more accessible to multidisciplinary audiences.

3.1.2 Frequency analysis

As previously mentioned, of the most basic yet useful analysis one can perform in a *corpus* of text is to understand how many times certain terms appear. While looking simple, the study of word frequency might be able to reveal important information.

To set an example, we'll use the *machado corpus* available in the NLTK python library and perform a few operations on it in order to gain insights regarding the text. This *corpus* is comprised of 2 books from acclaimed brazilian author Machado de Assis, *Memórias Póstumas de Brás Cubas* and *Dom Casmurro*. For this example, we'll use the latter.

A very simple analysis tells us the book has 82088 words, 9717 of them unique. Furthermore, if we run an analysis of concordance of the term *"Bentinho"*, one of the main characters in the book, we get the following list:

[1] We call a body of text a *corpus*, and when there are multiple bodies of texts, we have a *corpora*.

- *persiste na idéia de meter o nosso Bentinho no seminário ? É mais que tempo ,*
- *Não me parece bonito que o nosso Bentinho ande metido nos cantos com a filha*
- *Em segredinhos , sempre juntos . Bentinho quase não sai de lá . A pequena é*
- *faça desconfiar . Basta a idade ; Bentinho mal tem quinze anos . Capitu fez que*
- *Estávamos , sim , senhor ; mas Bentinho ri logo , não agüenta . Quando*

An analysis of concordance allows us to see more than the word alone: It displays contexts usually associated to that word, thus allowing to explore the text in a more in-depth manner than a simple word counting, for example.

Another possible way of doing so is to get contexts which appear frequently for a pair of words. As an example, we might be interested in knowing the contexts in which *"Escobar"* comes with *"Capitu"* since he is the main reason for the romantic doubt which torments Bentinho. A listing of that analysis returns the following:

- tinha é
- foi era
- refletiu de
- foi que interrompeu
- confessou e
- sorriu

The real list is bigger, but for the sake of brevity we'll only show the most relevant ones. If we wish to go further, we can also build a metric to analize how diverse a text is in terms of unique words in a giver corpus. Such delimitation can be defined as

lexical diversity = number of unique words/number of words in the corpus Dom Casmurro has a lexical diversity of 0.1183. As an example, *Moby Dick*, by author Herman Melville has a lexical diversity of 0.074, which only highlights the genius of Dom Casmurro's author Machado de Assis and the historical value of the text.

Another analysis which might be derived from the lexical diversity one is to see how frequently a term occurs during the text. If we plot the dispersion plot for the terms "Bentinho", "Escobar", "Capitu"and "Ezequiel", we get the following:

3.1.3 N-grams

Along with frequency-based techniques, n-grams are one of the most fundamental pieces which build up the NLP tool belt. Simply put, a n-gram is just a sequence of words,

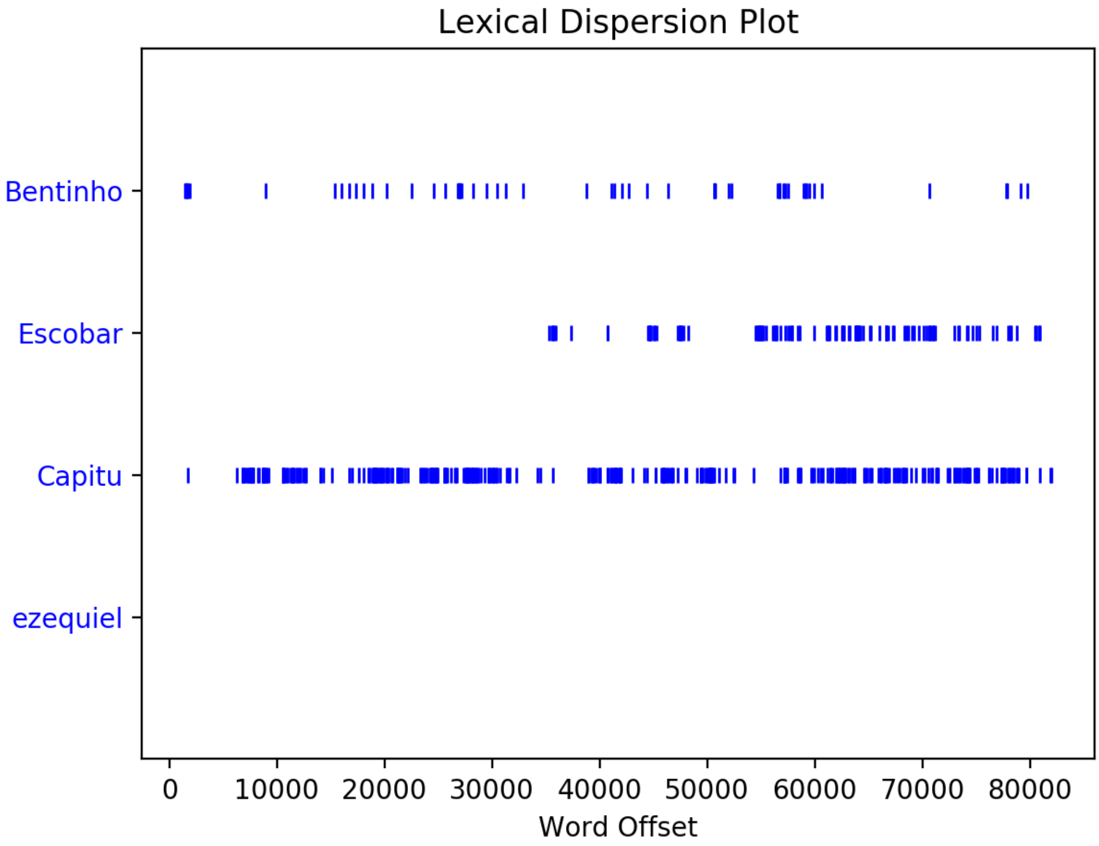


Figura 3.1: Lexical Dispersion plot

where "n"denotes the number of words that sequence has. For example, "New York"is a 2-gram, "She rises again"is a 3-gram ,and "The Winter is coming"is a 4-gram. "She rises again"is an example of a n-gram which doesn't happen in the language as often as the other examples.

One way of approaching n-grams is to think of them through a probability and statistics point-of-view. We can think, for example, in how the words tend to occur next to one another and how that translates to n-grams in a given language. For example, "Hot Chocolate"is much more likely to happen in a corpus of text written in English than "Hot cucumbers". Albeit being a simple idea, it already allows us to build prediction systems for entire languages and to begin to understand how they are formed. Another use of that idea is to build spelling-check applications, as "spilled water"tends to happen much more often in English than "spilled wter", for example. So if we're able o identify both expressions are formed mostly by the same letters and in the same order, but one happens much more frequently in the language than the other, we can safely derive the conclusion one might be just a spelling mistake instead of a real word or expression.

This idea is very important because through it we're able to build the **n-gram model**, which allows us to calculate the probability with which a word could happen based on the N-1 words that occur before it. So if we have a 2-gram, it means that we'll use the first word in order to predict the probability of the next one. Suppose we have the following corpus:

- Brasilia is a beautiful city
- São Paulo has an awful weather
- Brasilia is relatively close to São Paulo
- São José dos Campos is close to São Paulo

A real life corpus would be much larger and complex than this example, for the sake of brevity these four sentences will be enough to allow us to present the idea. Supposing we're using a 2-gram model in this example, in order to calculate the probability a word happening after another word, we'll rely on the concept of conditional probability, which states the following:

$$P(A) = (P(A) \cap (P(B))/P(B)$$

In this equation P(A) is the probability of a given event A occur, $P(A) \cap (P(B))$ is the probability of event B occurring given A has already taken place and P(B) of course, is the probability of event B happening. In our example, the probability of the word "is"to happen after "Brasilia"is equal to 1, given that there is only one occurrence of "is"and it is after "Brasilia". In contrast the probability of "Paulo"happening after "São"is 2/3, since the one other time "São"appears in our text is preceding "José", which has a probability of 1/3 of happening.

In the next sections we'll see how the concept of n-grams are one of the key ideas to Word2Vec, a powerful analysis technique which we'll use to analyze our data set searching for insights around the brazilian election of 2018.

3.1.4 Brief introduction to Artificial Intelligence and Neural Networks

Before we get to Word2Vec itself, we need to go through a very brief introduction to Artificial intelligence and Neural Networks, since Word2Vec works by using a 2-layered neural network in order to process text and derive conclusions around word proximity, for example.

Over the past few years, AI has gained the spotlight from various perspectives: Research conferences on developments of Artificial Intelligence are some of the most difficult to publish in, media has made thousands of articles debating new technologies and the ethical decisions behind them (with a special consideration to self-driving cars and their ethical duality between saving the driver or the people outside of the car in the case of an accident) and funding for AI-driven startups has reached skyhigh values, attracting a total of about $15.2B from 2013 - 2017[2]. With such high stakes, it is no wonder why the general public might still be confused on how AI works and what is the difference between AI and the more "traditional"computing.

From a very basic perspective, the key difference between AI-based programs and a non-AI one is that the usual approach for computer scientists and engineers to teach the computer to perform an operation is a usually very structured process. For example, if we need to teach a computer to bake a cake, the following steps might be taught to it:

Algorithm 1 Cake baking algorithm

```
procedure CAKE BAKING
    get ingredient white sugar
    get ingredient 2 eggs
    get ingredient 1/2 cup of milk
    get ingredient 1 1/2 cup of flour
    mix ingredients in a bowl
    put the mix in the oven for 30 minutes
    let the cake cool down for 30 minutes
    serve the cake
```

As you can se we teach the computer to expect a few inputs, process those inputs and output an answer, the cake. It is a well defined structured procedure, which leaves no ambiguity as to what we should expect when the outcome is produced. This has been the most popular way of writing computer programs ever since the computer was invented. We give it a raw input, teach it how to work with that input and expect it to deliver us a correct output.

With AI programs, however, this order is changed in a subtle but fundamental manner: Instead of teaching the computer the exact steps it should take in order to deliver a result, we provide both the input data and the results expected to the computer and let it derive the procedure. With this arrangement, we don't need to know the relationship between two entities ahead of time: We're able to let the computer figure out what patterns underline both groups and task it with giving new results for data inputs we didn't have before. This is, for example, the basic line for all recommender systems. When you log

[2] According to https://www.statista.com/statistics/621468/worldwide-artificial-intelligence-startup-company-funding-by-year/, accessed on April 1st, 2019.

into Netflix, or Spotify, or any service which recommends you to try out a new product, be it a song or a tv show, what it is actually doing is learning which products other people with a profile similar to yours like and offer them to you. In a summary, it learns from past actions of other users and produces a new output without having to be uniquely programmed to generate a feed of recommendations for any one specific user. Such learning can be categorized into two different categories: **Supervised learning** and **Unsupervised learning**. The first refers to programs which can expect to receive both the input data and the desired output for that input data. It then is able to generalize the processing it did in order to fit one group to the other to any new data of the same kind that comes its way. One of the most popular examples of such scenario is the MNIST dataset, which contains thousands of records of handwritten characters pictures along with a tag saying which characters they're in the english alphabet. The computer is then able to generalize for new images which characters they contain based on the training it received beforehand.

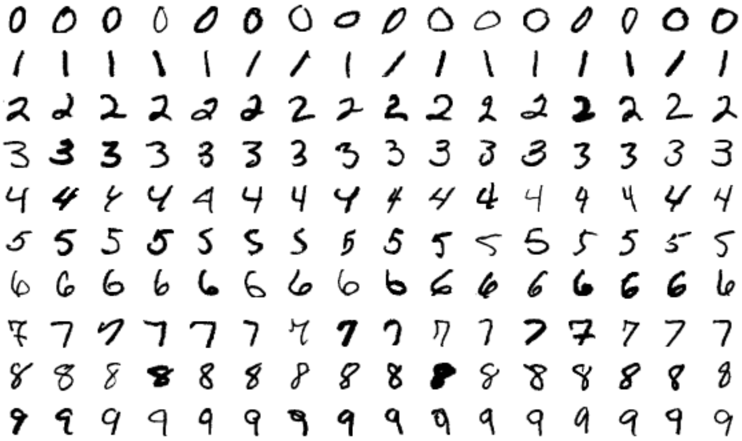


Figura 3.2: MNIST data set sample[3]

The other approach aforementioned, unsupervised learning, refers to algorithms which don't presume they'll receive a label associated to each instance of their training set and so they don't have a way to know the right answers. Such algorithms are very useful because they tend to be effective in showing us the inherent structure of the data they're receiving without having to make judgments as to whether the results they're producing are correct or not. One use example of such algorithms are **clustering** algorithms, in which we seek to discover groups of behavior within the dataset.

Another class of algorithms which are fundamental to our analysis are **Neural Networks**. They're defined by [15] as:

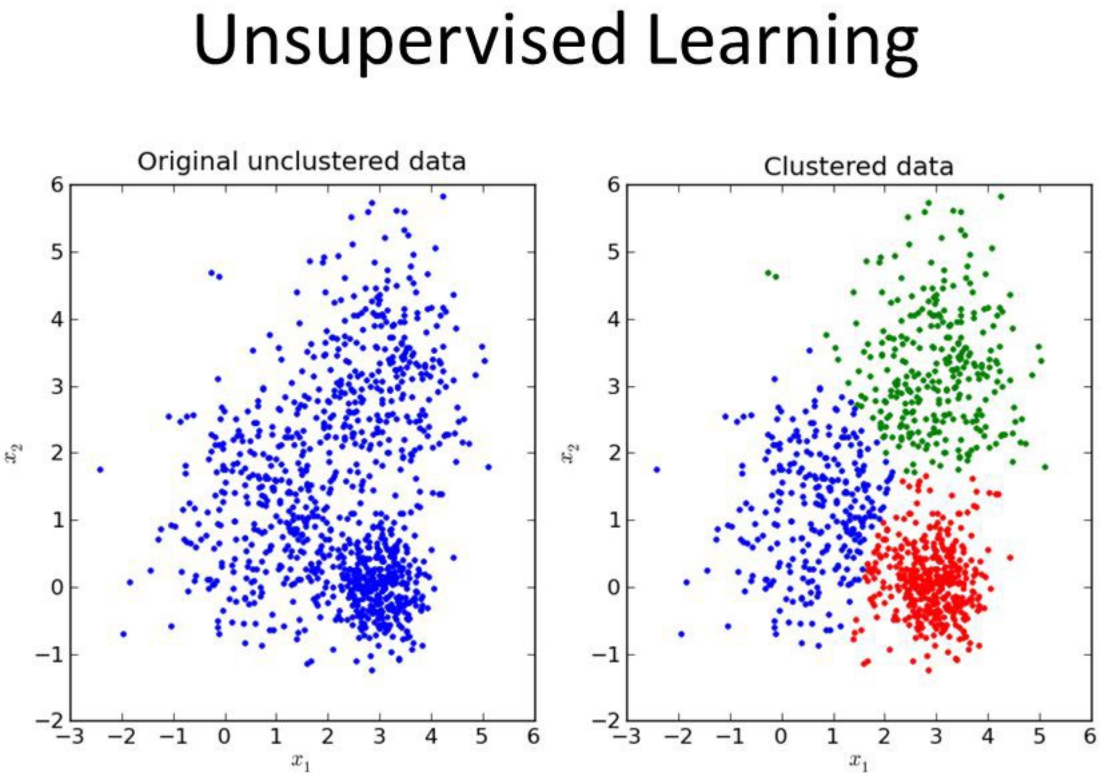


Figura 3.3: Unsupervised learning example
source: https://medium.com/the-21st-century/machine-learning-a-strategy-to-learn-and-understand-chapter-3-9daaad4afc55, accessed on April 3rd, 2019

> *"Neural networks are a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns. They interpret sensory data through a kind of machine perception, labeling or clustering raw input. The patterns they recognize are numerical, contained in vectors, into which all real-world data, be it images, sound, text or time series, must be translated."*[15]

Neural networks are a very important class of AI algorithms because they allow systems to be trained to classify data using multiple aspects of the dataset, called features. So for a group of images, for example, we might a neural network might discover that black and white pictures containing balloons belong to a certain group, black and white pictures which don't contain balloons to another and so on. Each of these characteristics might be recognized by one of the several layers of neurons of which a neural network might be composed of. Several times, neural networks are able to discover patterns in the data which aren't easily seen by humans, making them very efficient in providing us with insights which wouldn't be possible otherwise.

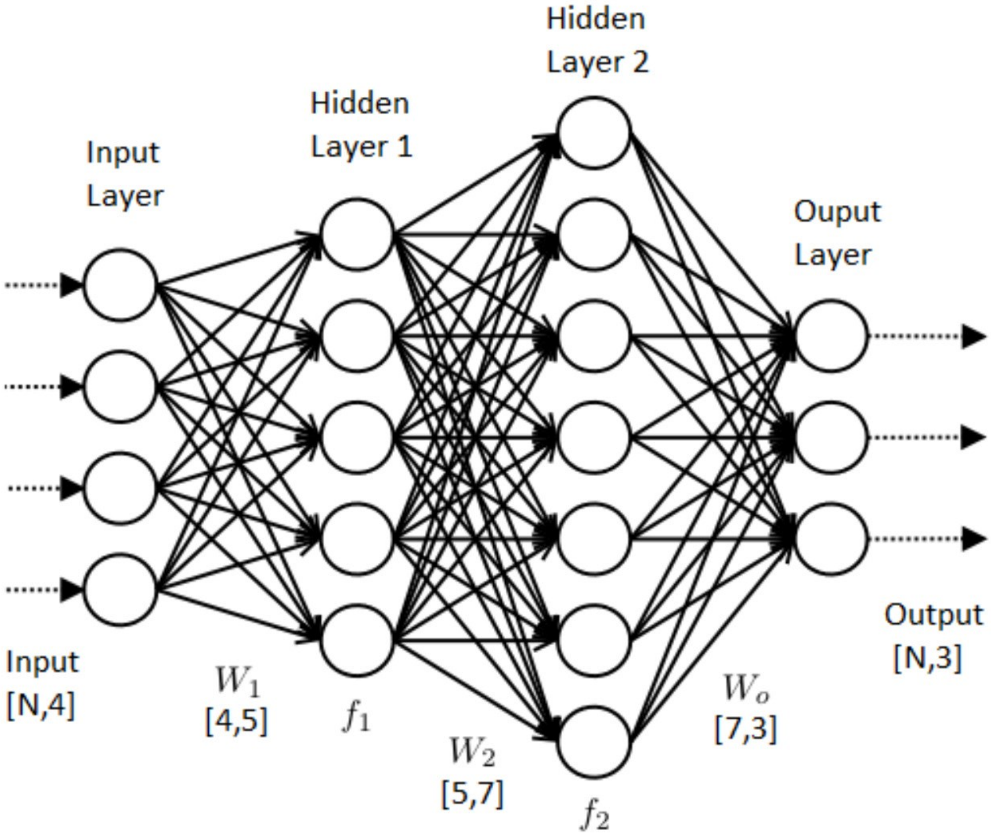


Figura 3.4: Neural Network example
source: https://medium.com/coinmonks/the-artificial-neural-networks-handbook-part-1-f9ceb0e376b4, accessed on April 3rd, 2019

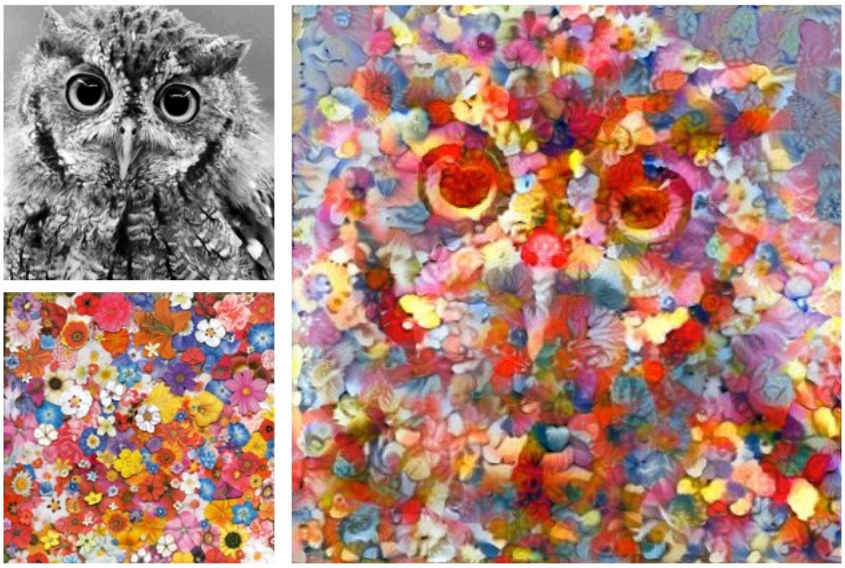


Figura 3.5: A very popular application of neural networks is the style transfer technique, in which the style of an artist is applied to another image, often resulting in a very interesting visual results. *source:* https://medium.com/data-science-group-iitr/artistic-style-transfer-with-convolutional-neural-network-7ce2476039fd, accessed on April 3rd, 2019

3.1.5 Word2Vec

A very important algorithm for us which builds on the ideas presented before, specially neural networks and frequency analysis, is the **Word2Vec** algorithm. So far the techniques we presented treat words as mostly units that are finished within themselves, therefore not taking into account, for example, notions of similarity between words. This is where Word2Vec excels: It allows us to construct semantic notions of a language by understanding which words occur next to one another and thus explore human communication in a much more sophisticated way.

It does so by leveraging techniques suited to learn high-quality word vectors from datasets with billions of words[16], an achievement which surpasses the previous state of art in a qualitative way: The more word vectors we're able to build from a corpus of text, the more intricate analysis we're able to grasp, since we're able to preserve the linear regularities between words.

The original paper which describes Word2Vec "presents two new architectural models for learning distributed representations of words that try to minimize computational complexity"[16]. The first of them is called **Continuous Bag-of-Words Model** and the second is known as the **Continuous Skip-gram Model**.

The **Continuous Bag-of-Words Model** builds on the idea of Feedforward Neural

Net Language Model. Two main layers build such a model: [4]: A *linear projection layer* and a *non-linear hidden layer*. Together, they're able to generate both a word vector representation and a statistical language model. In this context, a *linear projection layer* is responsible for mapping the indices of individual words in an n-gram context to a continuous vector space. It allows for even if a word appears multiple times during a text, each individual appearance of such a word still contributes to the weight that word will have on the projection layer output, the vector space containing the weights of all words. The following image might help the reader to understand the process more holistically:

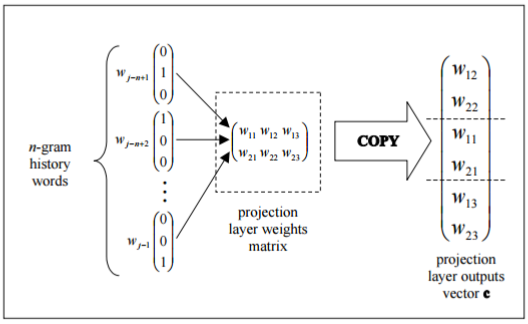

n-gram history words
w_{j-n+1}
w_{j-n+2}
w_{j-1}
w_{11} w_{12} w_{13}
w_{21} w_{22} w_{23}
projection layer weights matrix
COPY
w_{12}
w_{22}
w_{11}
w_{21}
w_{13}
w_{23}
projection layer outputs vector **c**

Figura 3.6: Schematics of the inner workings of Word2Vec. *source:* https:https://stackoverflow.com/questions/37889914/what-is-a-projection-layer-in-the-context-of-neural-networks, accessed on April 5th, 2019

In the picture, each neuron in the projection layer is represented by a number of weights equal to the size of the vocabulary. The sole objective of this layer is to project the given n-gram context, derived from the corpus, to a restricted continuous vector space which can be later processed by other layers in the classification task. The hidden layer is one of the layers responsible for categorizing and classification of words in the corpus, but it's inner workings have already been described in several other papers concerning neural networks, and furthermore, the Word2Vec model only uses one hidden layer in it's processing never building a full NNLM model, so we won't go into much detail here.[5]

According to the original text, the **Continuous Skip-gram Model** doesn't try to predict the current word based on context, and instead tries to maximize its classification by using another word in the same sentence. As they write, *"we use each current word as an input to a log-linear classifier with continuous projection layer, and predict words within a certain range before and after the current word. We found that increasing the range improves quality of the resulting word vectors, but it also increases the computational complexity. Since the more distant words are usually less related to the current word than*

[4] A very interesting brief introduction to Feedforward Neural Networks can be found at https://towardsdatascience.com/deep-learning-feedforward-neural-network-26a6705dbdc7

[5] We'd like to thank the Stack Overflow community for the answer which was the basis for this section: https://stackoverflow.com/questions/37889914/what-is-a-projection-layer-in-the-context-of-neural-networks

those close to it, we give less weight to the distant words by sampling less from those words in our training examples".

Hence, we see that Word2Vec differs from other textual processing algorithms in at least two fundamental principles: It doesn't try to understand words as entities which end in themselves (which is normally done in order to avoid processing costs) and it also uses both the context in which a word happens(via the Continuous Bag-of-Words model) for the current word and it also predicts the surroundings of a word following the Skip-gram model.

The result of such an architecture is that we're able to not only understand a word in her own, as a frequency analysis would limit us to do, but also to see the bigger picture of the role that word performs in the corpus. The most classic example of this idea is shown by [16] in the very beginning of the paper which describes Word2Vec. If you take vector "King", subtracts vector "Man"and adds vector "Woman", Word2Vec is able to deduce from the context of an English-speaking corpus of text that you're referring to a vector "Queen". Therefore, the nuances of gender and structural construct of the language are finally able to be understood in their whole, without us trying to teach the computer which grammar rules are important to a certain corpus of text or not.

In the next sections, we'll explore how we built a data set containing Google search results for multiple ideological profiles during the Brazilian Election period in 2018, as well as using Word2Vec to try to grasp some insights from that dataset.

Capítulo 4

Proposal

In this chapter, we're going to present a proposal on how to study the Brazilian general election of 2018 through the perspective of Natural Language Processing and the relationship between citizens and search engines, exploring how Google respond to users of multiple ideological positions. This chapter is a preview of sorts of the next chapter, where we indeed get our hands dirty and make a case study focusing on the election through the eyes of several personas we propose in this chapter.

4.0.1 Pre-analysis process

Our final objective is to be able to analyze in a macroscopic way how users from different ideological positions interacted and were affected (or not) by the results shown by Google during the election period. In order to do so, we first need to have data to work with.

Ideally, we'd be able to study real-world users who identify themselves with certain political positions in order to get a more accurate understanding of how real-world users use search engines in order to decide how to cast their votes. Being a small group of researchers, however, imposes certain constraints which we needed to adhere by, and a couple of them were financial limits and limited headcount.

4.0.2 Personas

To be able to continue with our study, therefore, we need a way to replicate, to the best of our efforts, the behavior of real users when interacting with search engines. To do so, we'll rely on developing several *personas*, which we'll appear to the search engine as a real user belonging to a certain political ideology when the time comes for us to begin the training phase, which we'll get into soon.

Next, we present a list of *personas*, their respective political leaning and the words we used during the *training phase*, which is an effort prior and during the election period to let Google understand how these accounts differ from one another, and how they relate to a user who doesn't perform political queries in the platform.

Three categories encompassed 15 *persona* accounts (N=15): [1]

- 6 profiles in the *ideological* category, evenly spread across right-wing and left-wing accounts,
- 6 profiles in the *gender* category, spread across men and woman,
- 3 neutral profiles, without search history to serve as a control group

The keywords used for each subgroup are as follows:

right-wing queries

imposto zero, direito armas, pena de morte, ideologia de genero, escola sem partido, intervencao militar, marxismo cultural, direito a vida, reducao maioridade, prisao perpetua, esquerdopatas, petralhas, estado minimo, reducao impostos, ditadura venezuela, ditadura cuba, feminazi, lula ladrao, privatiza tudo, liberdade economica

left-wing queries

lula livre, reforma agraria, direito a moradia,passe livre,diretas ja,fora,temer,volta dilma,coracao valente,quem matou marielle franco?,legalizacao do aborto,socialismo,feminismo,pre-sal e nosso,descriminalizacao das drogas,laicidade do Estado,anula STF,luta contra o racismo,contra o golpismo,defesa da universidade publica ,lute como uma menina

Man's queries

como fazer churrasco,tabela brasileirão,como dar um no de gravata?,roupas masculinas,câncer de próstata,saúde masculina,impotência sexual,viagra,paternidade acessórios masculinos,ejaculação precoce

Woman's queries

como amamentar?, roupas femininas, sintomas de menopausa, tensão pré-menstrual, cólica menstrual, maternidade, depressão pós-parto, câncer de mama ,câncer de útero, saúde feminina,acessórios femininos

[1] The accounts we created were the following: resocie.direita@gmailcom; resocie.direita2@gmailcom; resocie.direita3@gmailcom; resocie.esquerda@gmailcom; resocie.esquerda2@gmail.com; resocie.esquerda3@gmail.com; resocie.homem@gmail.com; resocie.homem2@gmail.com; resocie.homem3@gmailcom; resocie.mulher@gmail.com; resocie.mulher2@gmailcom; resocie.mulher3@gmail.com; resocie.neutra@gmail.com; resocie.neutra2@gmail.com; and resocie.neutra3@gmail.com

Although we did our best to select words and expressions that correctly represented each of our *persona* categories, we recognize that our choice of words might be subject to inherent biases and further research is necessary in order to determine the best approach in generating such terms.

In order to build the *ideological* category, we relied on the study[2] made public by the Brazilian Institute for Data Analysis in which they analyzed public pages on Facebook and found two poles: Pages which politically leaned to the left and to the right. After delimiting both of them, we ordered in descending fashion the pages by their centrality within the network, which we measured by the number of likes they had and the number of activities they performed. We also disregarded pages related to pre-campaign activities and political parties pages, partly because of the difficulty in fitting them under our simplified model of the political spectrum. It is also important to say that for accounts in the *ideological* category, no gender is specified and all of them had 40 years old as default age configuration.

4.0.3 Training phase

With the accounts created, we proceed to the next phase: The *training phase*. In this process, we'll try to teach Google how each account behaves. To do so we'll employ the following algorithm to access the search engine with each account, perform the search queries and log out, in a way that one account doesn't affect the other.

Algorithm 2 Google training

```
procedure GOOGLE TRAINING
    for user-account in personas:
    Open Selenium[3] and access https://accounts.google.com:
    For each search query:
    Type search query into search box
    Wait for the result, save a print-screen image and download the corresponding
  HTML
    For each URL in the result page:
    Access the URL
    Wait for the result and save a print-screen image
    Logout of the account
```

The frequency with which each account was accessed is determined by the following method: We choose a random number n and calculated the modulus of such a number by 3600. The result of this operation represents the number of seconds the computer waits until repeating the training procedure.

[2] The study is available here: https://s3.ibpad.com.br/redes/direita-esquerda/

4.0.4 Data collection phase

The data collection is similar to the training phase in the sense that it also interact directly with Google in order to work. The intent of this procedure is to collect the data Google provides the personas when they perform each search query. The algorithm used for such procedure follows:

Algorithm 3 Google data collection

```
procedure GOOGLE DATA COLLECTION
    for user-account in personas:
    Open Selenium[4] and access https://accounts.google.com:
    For each search query:
    Type search query into search box
    Wait for the result, save the relevant fields to database
    Logout of the account
```

The algorithms for the training phase and the data collection phase are similar, but they differ in that the training phase saves images later used to assess the quality of the training process. In contrast, the data collection phase saves the results directly to the database. The fields captured by the data collection procedure are as follows:

4.0.5 Working with the data

The database generated by the data collection phase unlocks many possibilities of ways one can undergo in the effort of understanding the Brazilian elections through the eyes of data analysis. We'll choose to apply the Word2Vec algorithm to different subsets of our dataset. As an example, applying Word2Vec with the titles of the results shown by Google lead to a more concentrated understanding of the results, since it leads to a small dataset. In opposition, if we apply it to the contents of the pages returned as results by Google we get a more well-rounded analysis, since our dataset is larger.

4.0.6 Word2Vec analysis of a body of text

The first application which we'll make use of Word2Vec is to the news' titles in our database. The reasoning for such an analysis is that it has been shown that users tend to click on the top results shown by a search engine[17] and therefore an analysis showing the proximity of words in that section might reveal interesting patterns. We then apply Word2Vec to the textual contents of the URLs returned by Google during our data collection phase.

The parameters we show in the algorithm above are the following:

Algorithm 4 Applies Word2Vec to a corpus of text

```
procedure Word2Vec
    $bodyOfText \leftarrow loadBodyText().lower().removeAccents().applyIramuteqFilter()$
    $wordTokens \leftarrow nltk.word\_tokenize(bodyOfText, language = "portuguese")$
    $word2vec\_holder \leftarrow Word2Vec(wordTokens, size = 10, min\_count = 2)$
    $word2vec\_holder.train(wordTokens, total\_examples = len(wordTokens), epochs = epochs)$
    $print(word2vec\_holder.wv.most\_similar('some\_query\_term', topn = 50))$
```

size: The size of the vector representing the context (or neighborhood) of each word.

window: The distance between a word and a neighbouring word. If the neighbouring word lives somewhere further in the text than the window limit, to the left of the target word or the right, then it won't be considered as related to that word.

min_count: The least amount of times a word has to occur in the corpus in order to be considered by the model.

workers: The number of threads the program is allowed to use in order to carry processing load.

The same procedure is used in order to analyze the news' titles and the content of the URLs we parse from our database, so in order to refrain from duplicate content in this document we won't reproduce the other algorithms, since the difference is mainly the input dataset.

Capítulo 5

Case Study

In this section, we'll apply the methodology we explained in the previous section to a real world scenario: Studying how Word2Vec can be used in order to understand the Brazilian general election of 2018 through the lenses of the interaction between users and search engines. We'll go through the steps of getting the necessary data and apply the algorithm to different parts of the dataset in order to try to extract the best possible insights.

5.1 Acquiring data

Multiple ideas arise when thinking on how to observe the interaction between users and search engines, specially when such interaction might influence power structures within society. To understand the impact of such technologies in user's day-to-day life three approaches are suggested, each with their own positive and negative sides. We'll briefly cover the first two, given that they were the basis for our final research design and weren't deployed *per se*.

The first idea was very *concierge*[1] inclined - We'd go to busy streets ourselves and ask people passing by to run a few queries on Google for us and email the research team a printscreen of the results, which would later become the primary source of data for the project. The upside of this approach is that it allows for a very randomized sample of voters. Standing on popular locations, we'd be able to talk to a multitude of social and economical landscapes, and gather data from a very diverse set of realities. However, it also means the amount of data collected would be linearly related to the number of people interviewed, which directly depends on the amount of researchers available to conduct such interviews. Since we're a small team, such limitation would fundamentally harm the outcome of our efforts.

[1] In the *startup* community, a concierge happens when a company provides a service in the most analog way possible, sometimes without involving technology at all.

Another possibility dealt with a controlled-environment data collection from a group of people. We'd invite about 40 people, encompassing students, staff and faculty of the university in order to run queries on Google while logged in to their accounts. The results would be stored, made anonymous and later studied by us. Although such approach would lower the number of researchers needed to conduct the experiment, it'd also create an inherent bias given the nature of subjects available to us - mostly well educated, financially privileged people from middle or upper social classes. The third approach - chosen as our *modus operandi* - is described below.

The remainder of this chapter describes the methodology chosen to run customized data-collection algorithms on accounts which aim to model the multitude of possible political leanings by Google's average user in terms of several *personas*[2]. Our research is composed of two phases: The *training* phase and the *data collection* phase. Both are necessary for us to teach Google the behavior of each *persona* account we created and then for us to learn how it reacts in an election setting to different searches from different *personas*, and derive a conclusion on whether or not the political leaning of a user accounts affects the way results are shown.

The first challenge faced by our research team was on how to determine which *personas* should be created in order to replicate the most accurate representation of real users with unique fittings in the political spectrum. We used several different accounts created solely for the purpose of this research, thus not inheriting any previous search history - we'll specify the details of such accounts in the development of this chapter.

Following, we needed to perform searches tailored to each of those users during the period of political campaign that preceded the Brazilian General Election in 2018[3]. We maintained regular queries during the election period, which allowed us to gather a dataset of about 380 thousand records, which are further described in the next sections.

Before we can discuss the technical side of scrapping data from Google, we needed to design a system able to make itself look like various users to the search engine. Even though it may sound as a simple task, triviality is as far a concept for us as possible - to fool the largest search engine available can hardly be classified as an easy assignment. Ad-based revenue models lead search engines - and obviously Google - to employ multiple techniques in order to avoid ad-related frauds[18][19]. However, in order to achieve meaningful results in our research goal, we need to be able to gauge how the mechanism

[2] A *persona* is considered here to be an analogy to a real person, emulating real searches performed on search engines

[3] Brazil's Tribunal for electoral purposes, the *Tribunal Superior Eleitoral (TSE)* specifies a period of 45 days prior to election day in which politicians and political parties are allowed to actively make use of different advertisement methods in order to reach their target audience

responds to a brand new user as he or she interacts with the platform for a certain amount of time.

Since there is no dataset available that could provide us with a mapping of political preferences to search history for a large enough amount of users, we needed to create our own avatars (the *personas* hitherto mentioned) that portrayed with reasonable fidelity real users and their search queries on Google, for a multitude of possible political opinions. A generalist approach, however, would increase complexity tremendously. As such,we decided to stick with a simpler approach.

Three categories encompassed 15 *persona* accounts (N=15): [4]

- 6 profiles in the *ideological* category, evenly spread across right-wing and left-wing accounts,
- 6 profiles in the *gender* category, spread across men and woman,
- 3 neutral profiles, without search history to serve as a control group

The keywords used for each subgroup are as follows:

right-wing queries

imposto zero, direito armas, pena de morte, ideologia de genero, escola sem partido, intervencao militar, marxismo cultural, direito a vida, reducao maioridade, prisao perpetua, esquerdopatas, petralhas, estado minimo, reducao impostos, ditadura venezuela, ditadura cuba, feminazi, lula ladrao, privatiza tudo, liberdade economica

left-wing queries

lula livre, reforma agraria, direito a moradia,passe livre,diretas ja,fora,temer,volta dilma,coracao valente,quem matou marielle franco?,legalizacao do aborto,socialismo,feminismo,pre-sal e nosso,descriminalizacao das drogas,laicidade do Estado,anula STF,luta contra o racismo,contra o golpismo,defesa da universidade publica ,lute como uma menina

Man's queries

como fazer churrasco,tabela brasileirão,como dar um no de gravata?,roupas masculinas,câncer de próstata,saúde masculina,impotência sexual,viagra,paternidade acessórios masculinos,ejaculação precoce

[4] The accounts we created were the following: resocie.direita@gmailcom; resocie.direita2@gmailcom; resocie.direita3@gmailcom; resocie.esquerda@gmailcom; resocie.esquerda2@gmail.com; resocie.esquerda3@gmail.com; resocie.homem@gmail.com; resocie.homem2@gmail.com; resocie.homem3@gmailcom; resocie.mulher@gmail.com; resocie.mulher2@gmailcom; resocie.mulher3@gmail.com; resocie.neutra@gmail.com; resocie.neutra2@gmail.com; and resocie.neutra3@gmail.com

Woman's queries

como amamentar?, roupas femininas, sintomas de menopausa, tensão pré-menstrual, cólica menstrual, maternidade, depressão pós-parto, câncer de mama ,câncer de útero, saúde feminina,acessórios femininos

While we tried to form coherent word clusters which represented real world users to best of our abilities, the question of how to determine the optimal group of words for each of the above categories remains open. Despite acknowledging that the words proposed by us might not fully grasp the complexity of dynamic social groups, we do believe that they're sufficient to our exploratory study. It is also important to say that the neutral accounts didn't receive the same training procedure the other accounts went through.

In order to build the *ideological* category, we relied on the study[5] made public by the Brazilian Institute for Data Analysis in which they analyzed public pages on Facebook and found two poles: Pages which politically leaned to the left and to the right. After delimiting both of them, we ordered in descending fashion the pages by their centrality within the network, which we measured by the number of likes they had and the number of activities they performed. We also disregarded pages related to pre-campaign activities and political parties pages, partly because of the difficulty in fitting them under our simplified model of the political spectrum. It is also important to say that for accounts in the *ideological* category, no gender was specified and all of them had 40 years old as default age configuration.

The preliminary step we haven't got into yet is that we made sure to set up computers specifically for the task of running our algorithms, doing our best to prevent Google from acquiring data that was on disk previously to the beginning of the research effort. Therefore, we formatted every computer and made sure they were exclusively operating for our research purpose, never allowing any other user to log into their Google accounts - or into any online service whatsoever - in order for our work to remain as pristine as possible.

In the next section, we'll address both phases of our research design and how they interacted with each other in order to produce as coherent of a dataset as possible.

5.1.1 Training phase

The training phase can be divided into two eras: The first, in which it was executed manually by a member of the research team, and the second, in which we developed an automate mechanism to make it easier to perform the search queries in each of the accounts we created before.

[5] The study is available here: https://s3.ibpad.com.br/redes/direita-esquerda/

The algorithm that powered the second phase is the following:

Algorithm 5 Google training

```
procedure GOOGLE TRAINING
    for user-account in personas:
    Open Selenium[6] and access https://accounts.google.com:
    For each search query:
    Type search query into search box
    Wait for the result, save a print-screen image and download the corresponding HTML
    For each URL in the result page:
    Access the URL
    Wait for the result and save a print-screen image
    Logout of the account
```

To determine the frequency with which we'd execute this algorithm in each of the machines available, we developed the following procedure: We chose a random number n and calculated the modulus of such a number by 3600. The result of this operation represents the number of seconds the computer waits until repeating the training procedure. With this procedure we were able to cover the whole electoral period and successfully perform the queries we needed.

5.1.2 Data collection phase

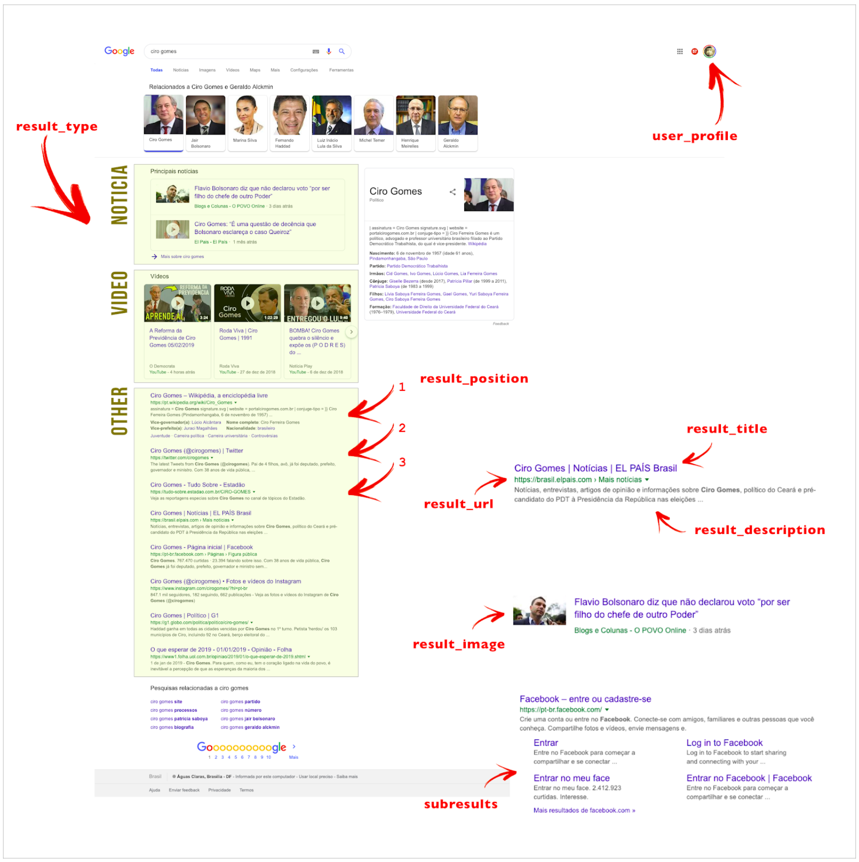
In order to run this phase, we partnered with a computer science professor at Universidade de Brasília, Cláudia Melo, who taught the Software Engineering course in the first semester of 2018[7]. Her class developed software to get data from multiple web services such as Twitter, Facebook, Instagram and some others. For this project, we're interested in the work done by the group responsible for monitoring Google.

The software developed to monitor search queries on Google is called 'Observatorio Google', and runs on Javascript libraries such as Node Js and Express JS. It's intent is to log into various Google accounts, perform search queries and store the results so that they're available on an easy-to-use manner.[8].

[7] We thank professor Claudia Melo, from the Computer Science department, and her Software Engineering class for her support in the realization of this research effort, including the students involved in developing the code for collecting data from Google: Douglas Alves Ferreira, Estéfane Helen, Gabriel Taumaturgo, Gabriel Almeida, Guilherme Castro, Juana, Léo Moraes da Silva, Lincoln Abreu Barbosa, Luis Braga, Luiz Filipe, Marcus Vinicius, Mikael Mello, Ricardo Rachaus, Tomas Rosário Rosemberg e Yan Trindade. Their work can be found in the following repository: < https://github.com/unb-cic-esw/Observatorio-google>.

[8] It is an opensource project, and information on how to install and use it is available at the following URL: https://github.com/unb-cic-esw/Observatorio-google/wiki/Realizando-coleta-dos-resultados-de-pesquisas-no-Google

The software was deployed and ran during the election campaign period, from August 13th, 2018 to October 30th, 2018. Several fields of data were stored, including the date and time in which the search was being run, as well as the *persona* account in which the software as logged in in order to run the query, the url of the first 26 results for each query on each account. The data stored from each result was the url to each result Google was pointing to, as well as the title and description of that specific result. We also stored metadata for image results, a short preview of the content of a website in case Google provided it and lastly, a boolean variable indicating if such result is an advertisement or not. The fields stored in our dataset are exemplified in the following image:



Ciro Gomes | Notícias | EL PAÍS Brasil
https://brasil.elpais.com › Mais notícias ▾
Notícias, entrevistas, artigos de opinião e informações sobre **Ciro Gomes**, político do Ceará e pré-candidato do PDT à Presidência da República nas eleições ...

Flavio Bolsonaro diz que não declarou voto "por ser filho do chefe de outro Poder"
Blogs e Colunas - O POVO Online · 3 dias atrás

Facebook – entre ou cadastre-se
https://pt-br.facebook.com/ ▾
Crie uma conta ou entre no **Facebook**. Conecte-se com amigos, familiares e outras pessoas que você conheça. Compartilhe fotos e vídeos, envie mensagens e.

Entrar
Entre no Facebook para começar a compartilhar e se conectar ...

Log in to Facebook
Log in to Facebook to start sharing and connecting with your ...

Entrar no meu face
Entrar no meu face. 2.412.923 curtidas. Interesse.

Entrar no Facebook | Facebook
Entre no Facebook para começar a compartilhar e se conectar ...

Mais resultados de facebook.com »

Figura 5.1: Google's fields captured in the data collection phase *source: Research group*

We performed 13564 queries during the period between August 17th, 2018 to October 30th, 2018, which resulted in 235.570 thousand lines in our database. Each line in the database is an observation which represents an URL in an specific result page, for each search query in each of the accounts we created. From the whole corpora of URLs, 8.883 of them were unique addresses. However, some portion of these observations presented all kinds of noise which we only realized later in our process. In further sections, we'll present the final number of rows we were able to work with. Among these, when we analyze only results referring to news and results (therefore not taking into account ads and videos), we identified 285 unique domain names, and from those, we plotted the 15 most accessed ones which represent 65% of all results:

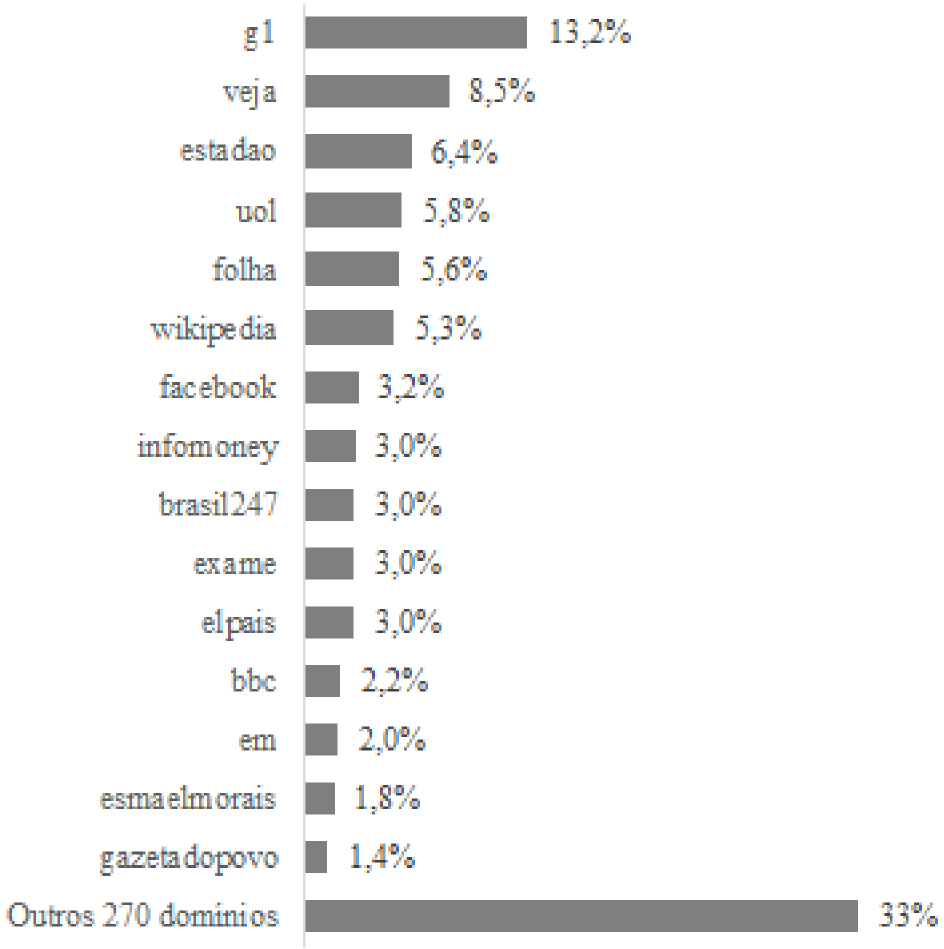


Figura 5.2: Google's fields captured in the data collection phase *source: Research group*

5.2 Understanding the data acquired

Now that we gathered a reasonable amount of records from our previous effort of acquiring data from Google, we're in the perfect place to grasp some meaning from it and try to understand the effect the usage of Google had on voters during the Brazilian Election period in 2018. There is a multitude of possible analysis which could lead to very interesting discoveries and insights. Of course, however, due to time limitations and resource constraints, in this paper we'll adhere to one line of thought for the sake of organization and efficiency.

5.2.1 Word2Vec analysis of news' titles

As we have previously mentioned before, Word2Vec is a great tool for developers and researchers to extract more meaning from bodies of text, because it is more robust and thorough than other analysis techniques. In the dataset we acquired and which was described in the previous section, one of the fields is intrinsically text-based(instead of being, for example, an structured URL field) and therefore suited for the use of the Word2Vec algorithm: The title of each result.

But before we go on to describe the implementation of our program, a previous step must be taken. The data acquired contains several lines of invalid data, as well as data from dates which are of no interest to us (e.g before the election period). Thus, we need to clean the database so it only contains lines which help in our analysis. The following algorithm was applied to the original database so it became more suited to our efforts:

Algorithm 6 DB cleaning Algorithm

```
procedure DB cleaning Algorithm(
    for row in database:
    if date is valid:
    if date is equal to or later than August 16th, 2018:
    if URL in $result_urlfieldisvalid :$    save to row to new file)
```

Now that we have a clean dataset to work with, we can proceed to apply Word2Vec to news' titles.

Once again, we'll describe step by step the steps taken to achieve the result. Among others, we import the Word2Vec module, which comes from the Gensim library[9] and will allow us to not have to implement Word2Vec by ourselves, an effort which could take weeks if not months of work to complete. After that, we import NLTK, a Python-based NLP library which provides us with several corpus of text if we need to (and we most probably will in the next section) and stopwords, which allows us to remove words that aren't necessary to our analysis from our body of text.

The next thing we do is to define a method remove_accents, which is necessary due to the difference of char sets between Portuguese and English. Since Portuguese is a language derived from Latin, there are many characters unfamiliar to English speakers, and in our case, libraries. Those characters may sometimes break our application, and thus we'll remove them beforehand.

After that, we build two arrays, which will hold the whole body of text for both left and right accounts, and proceed to store all news titles in those variables. After we're finished storing them in the arrays, we proceed to make a quick processing of making all

[9] Available at https://radimrehurek.com/gensim/

characters lowercase, substituting invalid characters for valid ones and so on, so that our application doesn't break for unexpected reasons.

We then start building the model for the Word2Vec algorithm. We tokenize words - which means we build a vector of weights for each word appearing in the text. After that we remove all stop words from that body of text and build our Word2Vec model. Finally, we print the most similar words to 'Haddad' and 'Bolsonaro', the two candidates for the presidency in the 2018 elections. We perform this procedure for both rightwing and leftwing accounts so we're able to take a full look on how both candidates relate to each political leaning.

5.2.2 Extending the analysis: Body of the news's articles

The analysis of the news' titles is certainly interesting and introduced us to several insights which allowed us to see further than a simple frequency analysis would. However, it is also limited by the scope of the body of text: Each news' title has only a few words, and even though we're working with a large dataset it can still lead to a body of text not large enough that we're able to extract the maximum from the Word2Vec horsepower.

For this reason, we'll proceed to build an even larger body of text. Two main options come to mind: To either use a ready-to-use corpus of text or to build or own. A tryout was carried with the first option in mind with the MacMorphus[10] corpus, which include over a million words from one of the largest news papers in Brazil, *Folha de São Paulo*. Even though it is a very interesting corpus of text to carry analysis on, there is a temporal problem which arises with it: According to the NLTK documentation, it is comprised of news text from 1994, which can't possibly mention any of our subjects of interest for obvious reasons. Hence, the idea of joining two corpus of text (our news' titles and the MacMorphus corpus) is not a good one: Since MacMorphus is so much larger in length than our corpus, it makes so our two political agents in focus (Jair Bolsonaro and Fernando Haddad) not to appear as frequent enough terms when we build the Word2Vec tokens and subsequently search for most similar terms. Therefore, we're left to only one option: to visit each of the urls corresponding to the titles we mentioned before, download their contents and build a Word2Vec mapping from that source.

5.2.3 Building a parser for news' urls

Before we're able to build a Word2Vec representation of a dataset we need, of course, to gather the data refering to that dataset.

[10] http://www.nltk.org/howto/portuguese$_{e}n.html$

In our case, several challenges arise when doing so: To build a webscraper to visit and extract text from one URL or several URLs, it is expected that we input the scraper with the structure it should be looking for. As an example, for a page with HTML code such as the following:

```html
<html>
<head>
<title>
An example page for this research effort
</title>
</head>
<body>
<div class="Example1">
Some cool text
</div>
<div class="Example2">
Other cool text
</div>
</body>
</html>
```

We can build a scraper, which will download the whole HTML file, and then a parser to look specifically for text inside the "Example1"div tag but not for "Example", and return us only the "Some cool text"content.

We're working with multiple pages from multiple domain names, however, and each one of them are structured differently, thus impeding us to proceed with our analysis if we're to try and code each page structure manually, since we're dealing with hundreds of different pages. In order to solve this problem, we'll adhere to one of the guidelines the Web has had since it's inception: Each fundamental HTML tag (<p>, <body>, <head> and so on) should be used for a certain reason. Thus, we'll capture the all <p> elements in each page. Of course we might get some junk content which isn't expected along with valid data, but it shouldn't be too much of a hassle since the Word2Vec algorithm will filter out words that don't appear too frequently in our combined body of text. Specific unwanted text such as Ads, for example, won't influence the whole of our analysis.

5.2.4 Scrapping data from news' URLs

In our study, we need to visit a few thousand addresses and save as much text as possible from each page so we end up with a large enough corpus of text which won't lead us to overfitting[11] when training the neural networks which power the Word2Vec algorithm.

[11] Overfitting, in the context of Machine learning algorithms, happen when an algorithm is over trained with a set of data so much that it loses the ability to generalize it's patterns to incoming new data.

Since our dataset contains multiple domain names, ideally we'd fetch each of them, understand their structure and then query the page so we could precisely extract the text needed. However, the amount of addresses we need to visit and the variety of their top-level addresses (which encompass multiple media outlets, Youtube, personal blogs and many other) make it impossible for us to hard code their structure into our program without spending time and effort which we can't currently afford. Therefore, a simple solution is used: we query each page, parse it's HTML content with Python library BeautifulSoup[12] and extract only <p> tags from them. This tag was originally developed to be used by paragraph elements in HTML pages, and we can reasonably assume most text in webpages still occur inside such tags[13]. Hence, our procedure was to open the file containing all news' titles and URLs and for each of them we visited the respective pages and downloaded all text inside <p> tags, storing them in a CSV file containing the mapping between the URL visited and the respective text.Also, in discussing this work with fellow researchers from our group, it was raised the issue of collecting data from <p> tags to be a poor heuristic to base our work upon. With this in mind, we built a small program which showed us which were the 30 most frequent websites in our database[14]. We then visited manually each of them and discovered that 29 of them stored their textual content inside <p> tags. Of course, we weren't able to visit some URLs due to issues like timeouts, servers resetting the connection and pages not being available. We saved a list of URLs unavailable to us and made it available online[15]. Please note that some domains blocked our program for every sub domain the owned. We were later able to access some of those links by changing our User-Agent configuration[16], which mimicks a real world user agent. Only after that we were able to bypass some filtering agents who were programmed to think of our access as being a robot.[17]

After we deployed our scrapper and let it run for a few hours, we were finally able to gather the data we needed. Overall, we visited 5199 unique URLs, with 59 of them returning null results and being discarded from our analysis. From that procedure, we were able to build a corpus of text consisting of 3.096.474 total words, with 174.226 being unique words. Also, we applied a stopwords filter built by us based on the stopwords

[12] Further reading available at https://www.crummy.com/software/BeautifulSoup/bs4/doc/

[13] Wikipedia, for example, stores all texts for articles inside <p> tags as of the time of this writing

[14] We've only considered the base url, such as 'http://g1.globo.com' in this counting

[15] Link to access: https://gist.github.com/teogenesmoura/741d31b11891aefbfd1892998265639d

[16] We used the following configuration:*'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X* $10_1 2_2) AppleWebKit/537.36(KHTML, likeGecko)Chrome/55.0.2883.95Safari/537.36'$

[17] It is important to mention that our algorithm didn't impose any load a webserver wouldn't receive normally: We queried for a few webpages only a few times a day, for a couple of days only, therefore not performing any kind of harmful behaviour with our program.

specified by the NTLK library[18]. The decision not to use the NTLK implementation of stopwords checking stems from the fact NLTK's implementation is based on a list[19], which is a data structure with a time complexity of O(N) for search operations[20]. Such complexity leads to a tremendous amount of time being wasted by the algorithm on large datasets such as ours, which took a few hours to filter all stopwords in a regular computer. Therefore, we decided to implement the stopwords filter based on a Set data structure instead of using the built-in function. A set is a data structure with time complexity of O(1) for most operations, including search, and we were able to deploy our algorithm almost instantaneously with a set containing all stopwords. On top of that, we also applied the Iramuteq filter, a lexicon database for words in portuguese which allow us not only to reduce the noise in our dataset by excluding words that don't make sense as well as by knowing the grammatical classification of the words in our dataset.

We also captured URL's content for each account in the ideological category using the same procedure as before, only with a simple conditional statement to select which records should be included. After applying the stopwords filter, we ended up with a dataset with 722042 words for right-wing accounts, with 66962 of them being unique, while for left-wing accounts we gathered 426835 words in total with 4805 unique ones. The difference in the volume is explained by the distribution of URLs in the file. In total, our algorithm saw 136794 urls, but only visited 5199 since they were the unique ones. Left-wing urls consisted of a body of 731 websites, while right-wing ones accounted for 1275 addresses.

5.2.5 Applying Word2Vec to each dataset

After scrapping and processing the URLs from the database, we found ourselves with three bodies of text: the *general* one, containing the text for every unique page in our original database, the *left-wing* one, containing text for left-wing accounts and the *right-wing* one. These bodies of text consist each in text files containing all <p> elements extracted from each web page in sequential order by our scrapping algorithm. We're now able to execute the Word2Vec algorithm against each of them and perform several analysis. In order to do so, we'll demonstrate the whole procedure used to perform that operation and then explain the parameters the Word2Vec algorithm accepts as inputs and in the next section we'll show the results of each iteration of our program.

[18] The full list of stopwords is available at https://github.com/xiamx/node-nltk-stopwords/blob/master/data/stopwords/portuguese

[19] Which can be seen here: https://bit.ly/2P3L1Et, from line 247 and onward

[20] according to Python's documentation available at https://www.ics.uci.edu/ pattis/ICS-33/lectures/complexitypython.txt

The procedure which applies the Word2Vec algorithm to each of our bodies of text is demonstrated below. We'll only show one instance of it because it is the same procedure, with the only variance being the input text file.

Algorithm 7 Applies Word2Vec to a corpus of text

```
procedure WORD2VEC
    $bodyOfText \leftarrow loadBodyText().lower().removeAccents().applyIramuteqFilter()$
    $wordTokens \leftarrow nltk.word\_tokenize(bodyOfText, language = "portuguese")$
    $word2vec\_holder \leftarrow Word2Vec(wordTokens, size = 10, min\_count = 2)$
    $word2vec\_holder.train(wordTokens, total\_examples = len(wordTokens), epochs = epochs)$
    $print(word2vec\_holder.wv.most\_similar('some\_query\_term', topn = 50))$
```

In this procedure, we follow a few steps which are now presented in greater detail. The first one loads the corpus from a text file and applies a two filters to make it more suitable to later processing, removing unwanted characters and making every word lowercase. Then we call the *word_tokenize* method from the NLTK library, which uses the Punkt sentence tokenization models[21] to parse strings into substrings[22].

In sequence, we build the Word2Vec model. This method allows for the following parameters:

size: The size of the vector representing the context (or neighborhood) of each word.

window: The distance between a word and a neighbouring word. If the neighbouring word lives somewhere further in the text than the window limit, to the left of the target word or the right, then it won't be considered as related to that word.

min_count: The least amount of times a word has to occur in the corpus in order to be considered by the model.

workers: The number of threads the program is allowed to use in order to carry processing load.

In the next section, we'll play around with some of these values and see differences in results. Next,we train the models considering the corpus of text in its entirety. The interesting parameter in this method is the *epochs* one. It defines the number of times the corpus of text will be iterated by the algorithm[23]. In general, the more iterations the better but with caution not to create problems such as over or underfitting.

Lastly, we print the most similar words to a given term and the number of words we want to see. This is just an example of analysis made possible by Word2Vec, and we'll explore some more in the results section.

[21] Further explanation can be found at https://www.nltk.org/$_modules/nltk/tokenize/punkt.html$

[22] An example to illustrate this is found at https://www.nltk.org/$_modules/nltk/tokenize.html$

[23] According to https://radimrehurek.com/gensim/models/word2vec.html

In the next section we'll show the results the Word2Vec analysis for each dataset we collected, the one containing texts for all accounts in our database, the one containing only text seen by right-wing users and the one with text seen by left-wing users.

Capítulo 6

Results

In previous sections we've introduced the context in which the technical analysis present in this paper exists: An evolving 21st century western society which sees itself building innovative technology and at the same time having its social interactions shaped by technology. In this section we'll observe some of the results of applying the Word2Vec algorithm to a dataset containing approximately a hundred thousand observations of Google search results for users of different categories, with differences by gender and political leaning, for example.

One important point we need to make before we can proceed to the discussion of the results is that the Word2Vec algorithm carries a certain inherent randomness built into it, which means that a person running the same program we used, with the same parameters we used for the same input dataset might see different results from the ones we see here. One possible reason for that effect might stem from the size of our datasets: Even though the largest of our datasets contains more than 2 million words, the smallest dataset with which the original Word2Vec paper[16] tests the algorithms contains 24 million words and best results are shown for datasets with 6 billion words or more. Therefore, we will consider our results from the standpoint of an exploratory analysis and keep in mind that better results would come up with a larger amount of data and processing power.

6.0.1 Word proximity of news' titles

First, we'll approach the dataset consisting of news' titles and see which kind of results we're able to gather. Since this dataset is vastly smaller in terms of quantity of words than the one containing the content of each page from the results, we'll use $size = 10$, $min\_count = 4$ and $epochs = 35$. The results we find with such values are:

Words similar to "Bolsonaro"in left-wing accounts:

Word	Coefficient
elaborar	0.8600923418998718
temer	0.8334468603134155
conta	0.9978168606758118
enxuta	0.9954599142074585
creches	0.9945040345191956
comentar	0.9943751096725464
neles	0.993672251701355
tributos	0.9914667010307312
vomito	0.9909074306488037
ibope	0.9904252290725708

Words similar to "Haddad"in left-wing accounts:

Word	Coefficient
comunista	0.9993612766265869
institutos	0.9992505311965942
abril	0.9991971254348755
politicas	0.998961687088012
agressoes	0.9976707100868225
medo	0.9972601532936096
usou	0.997238278388977
convictos	0.9971490502357483
confunde	0.9968955516815186

Words similar to "Bolsonaro"in right-wing accounts:

Word	Coefficient
presidenciavel	0.9997791051864624
odebrecht	0.9989746809005737
processo	0.998806893825531
unir	0.9984559416770935
discutir	0.9984230399131775
assistir	0.9979907870292664
jn	0.9979436993598938
define	9972240924835205
redu	0.9971490502357483
revela	0.9968955516815186

Words similar to "Haddad"in right-wing accounts:

Word	Coefficient
armou	0.9997791051864624
jato	0.9989746809005737
nesta	0.998806893825531
pontes	0.9984559416770935
fachin	0.9984230399131775
perder	0.9979907870292664
roteiro	0.9979436993598938
cartilha	9972240924835205
jaques	0.9984614849090576
robos	0.9984855651855469

From this first analysis, we can already uncover some discoveries that might help us understand the context in which the Brazilian Elections of 2018 happened. First, we see a clear divide in respect to how the candidates are characterized in their respective political niches. While Fernando Haddad is associated to terms such as "medo"(*fear*), "agressoes"(*aggressions*) and "comunista"(*communist*), all terms that induce negativity (we should remember that one of the key mottos of Bolsonaro's campaign was the fight towards the so called communists, left-wing politicians and political parties that were positioned against his campaign for the office), Jair Bolsonaro, to right-wing accounts, is more likely to be found close to terms such as "presidenciavel"(*presidential*), "odebrecht"(The Brazilian construction company associated with corruption scandals which involved Fernando Haddad's party, PT), "unir"(*unite*), "discutir"(*to discuss*), and "jn", which refers to the most viewed journalistic daily TV show in the country, in which he was invited to discuss his proposals as a candidate. Therefore, we can clearly see that Fernando Haddad's results are much more likely to have a negative construction towards them than Bolsonaro's one. This isn't unexpected: The 2018's election is already regarded as one of the most polarized elections of Brazil's history, and Bolsonaro's campaign strategy of putting his adversary's party corruption scandals as something he is the clear opposite seems to have worked on Google's search results.

When we look at the candidates in the light of their adversarial voters, we see a result which can be describe as expected. Haddad is highly mentioned to terms such as "Lava Jato", the judicial operation which investigated corruption schemes during the presidency of Luiz Inácio Lula da Silva, former president of the country and political partisan of Fernando's, "perder"(*lose*) and "Fachin", a Brazilian Judge in the Supreme Federal Tribunal who refused attempts made by Lula of getting an Habeas Corpus. In the other hand, for left-wing voters, "Bolsonaro"is associated to "UTI"(*ICU*), which seems to refer to the period he spent in the hospital after he was stabbed by a man while campaigning[1] , "roubou"(*robbed*) and "vomito"(*vomit*).

One might argue, however, that our analysis can't be taken as too profound or comprehensive since our coefficient values are always very close to 1, which might indicate some form of overfitting of the data available in relation to our model. For this reason, in the next subsection, we'll explore our larger dataset which consists in more than 2 million words extracted from the HTML content of pages which were visited by our algorithms during the election period.

[1] Further reading available at https://g1.globo.com/politica/noticia/2019/01/27/cronologia-atentado-contra-jair-bolsonaro.ghtml

6.0.2 Word proximity of news' contents

In the previous section, we showed the first 10 results in terms of proximity to "Bolsonaro"and "Haddad"for voters in left and right-wing accounts. However, that dataset might adversely affect our analysis since it is a small one, which leads our algorithm to overfit quite easily. In this section we'll show the results for the analysis of our larger dataset, consisting of 2.734.999 words, with 160.756 of them being unique. It is important to notice that we have almost 3 million words, this is still considered a small dataset in terms of big data. As explained previously, Word2Vec works better the larger the dataset is and ideally, we'd have a dataset consisting of billions of words.

We'll divide our analysis for this section in the following manner: First we'll explore the dataset as a whole and try to get some insights which aren't dependent on the category of the account being analyzed. Then, we'll analyze datasets which pertain to left-wing and right-wing accounts only. While they'll be necessarily smaller than the more general approach, we might be able to see tendencies unclear if we consider the whole spectrum of text. Without further ado, let's get to it!

6.0.3 Analyzing the News's contents

For this analysis, we'll apply the algorithm mentioned in the previous section to our body of text which is composed of the body of all URLs in our original dataset. It is important to remember that exceptions occurred: Some pages didn't store their main texts in <p> tags, others returned empty responses while others were plainly inaccessible by our script. We've logged the URLs we weren't able to access which can be found in Appendix 1.

Since in this part of the analysis we're not bound to profile categories, in the next table we'll provide words most related to both "Bolsonaro"and "Haddad". We've used several parameter changes to arrive at this list and we selected words that we believe contribute to understanding the 2018 elections and discarded some which at first glance we believed wouldn't be of much help. Since politics is a sensitive subject, we'll make the full list of words public for each candidate so that the interested reader is able to draw their own conclusions. Also, considering that the majority of websites we pulled content from are Portuguese-speaking, we won't consider words in English which may eventually find their ways to the most related to each candidate.

The parameters we used to build these lists were:

size = 5, min_count = 2, epochs = 30
size = 5, min_count = 2, epochs = 35
size = 5, min_count = 2, epochs = 40

Words similar to "Haddad":

Word	Coefficient
incompetente	0.9829217791557312
doleiro	0.9812902212142944
ideologia	0.9773783683776855
correio	0.9769439101219177
hebraico	0.9758862853050232
alencar	0.9449862241744995
maluf	0.9310760498046875
escolarizar	0.9265521764755249
doria	0.9257642030715942
corruptor	0.9225714206695557
enriquecer	0.9220681190490723
buarque	0.9192888140678406
estuprador	0.9115221500396729
diesel	0.9087313413619995
prepotente	0.908629655838012
disperso	0.907451152801513
preso	0.88321423530578
freixo	0.8770759105682373
gay	0.8667276501655579

Words similar to "Bolsonaro":

Word	Coefficient
descumprir	0.9965560436248779
capaz	0.9935079216957092
solidarizar	0.9902158379554749
inimigo	0.9768103361129761
dirceu	0.9737780094146729
kim	0.961829423904419
mente	0.9257142543792725
pragmatismo	0.92177653312683
indenizar	0.9182114005088806
nazista	0.916954517364502
rato	0.9145997166633606
museu	0.9032431840896606
gadelha	0.901979386806488
rouco	0.900751531124115
autoritarismo	0.8991959095001221
desequilibrado	0.896671712398529
constranger	0.8905130624771118
socialista	0.858917236328125
criminalidade	0.8540067076683044
louvor	0.8350745439529419
atrito	0.8333269357681274
fragilidade	0.8306527733802795
articular	0.8221142888069153
economia	0.8171266317367554
franco	0.8194248080253601)
radicalizar	0.8094037771224976
empresarial	0.7964493036270142

We believe both tables are able to summarize 2018's election in a few words. Several facets of it are represented by a few words. As an example, both candidates are associated to terms which are highly positive ,such as *solidarizar* for Bolsonaro and *escolarizar* for Haddad, as well as extremely violent ones, such as *estuprador*(rapist) for Haddad and *Nazi* for Bolsonaro. These relations shows us one aspect of the election: The highly polarized debate that took place. Furthermore, it also shows what might be considered a display of populist discourse, with *inimigo*(enemy) and *radicalizar*(to be radical) appearing for

Bolsonaro[2]. Topics such as corruption (*Dirceu, doleiro, corruptor*), religion(*louvor*) and political allies and adversaries are also represented(*doria, dirceu, kim*). It also shows characteristics which would be important in the future: Bolsonaro is heavily related to *fragilidade*(fragility), *atrito*(attrition) and *articular*(referring to the ability to make political connections in order to govern). These are three key topics of his government a few months into it. His son Carlos Bolsonaro, who is also an elected representative, constantly gets into public arguments via social media with the vice-president Amilton Mourão [3]

Our dataset is also able to show more direct relations which enhance the notion of a highly polarized campaign for both sides. For example, if we use the *most_similar* method with the *positive* parameters being *'haddad'* and *'bolsonaro'* and the negative being *'louvor'*[4], the result is 'aula'(*a class*). This shows very clearly the difference in approach to the election by the two candidates: Bolsonaro is known to be heavily supported by religious authorities, specially from very powerful evangelical churches, while Haddad is a professor who relies on his position as a teacher at Universidade de São Paulo for entering discussions about education, for example.

[2] If we consider the usual structure of a populist discourse, with the components of a great leader, the 'others', who are seen as enemies of the people, and people themselves

[3] *https://www1.folha.uol.com.br/poder/2019/04/bolsonaro-diz-querer-colocar-ponto-final-na-briga-entre-carlos-e-mourao.shtml*

[4] A kind of religious ceremony common in evangelical churches

6.0.4 Left-wing news' content analysis

We'll repeat the components of the analysis we introduced for the general body of text. In this section and the next, we'll fist present the data than discuss the findings.

Words similar to "Bolsonaro"in left-wing accounts:

Word	Coefficient
projeto	0.905442476272583
jobim	0.8635315895080566
ironizar	0.8536474108695984
manipular	0.8408524990081787
teatro	0.8380136489868164
telejornal	0.8316670656204224
homenagear	0.8174992203712463
progresso	0.815588653087616
mole	0.8048369884490967
correio	.7827032804489136
indiciar	0.7788822054862976
universidade	0.7716076374053955
apanhar	0.7497464418411255
janaina	0.7411710023880005
record	0.7367274165153503
mandante	0.7304861545562744

Words similar to "Haddad"in left-wing accounts:

Word	Coefficient
levantar	0.9681728482246399
covarde	0.968074381351471
receio	0.9613124132156372
disseminar	0.9549825191
boneco	0.9529079794883728
propriedade	0.9372718334197998
aliar	0.919761061668396
despudor	0.9185368418693542
obra	0.9178704023361206
protesto	0.903983473777771
ministerial	0.8894245624542236

6.0.5 Right-wing news' content analysis

Words similar to "Haddad":

Word	Coefficient
tranquilao	0.986917197704315
ironia	0.9606291055679321
hitler	0.9568542242050171
hipocrisia	0.9329890608787537
mamar	0.8936359286308289
ensino	0.8795946836471558
saneamento	0.8717922568321228
renovar	0.8519589304924011)
assertivo	0.84242206811904

Words similar to "Bolsonaro"in right-wing accounts:

Word	Coefficient
presidenciavel	0.9997791051864624
odebrecht	0.9989746809005737
processo	0.998806893825531
unir	0.9984559416770935
privatizar	0.8663020133972168
caos	0.8695659637451172
escritor	0.8522520065307617
mito	0.865236222743988
desenvolvimentista	0.8203554749488831
algoritmo	0.8258252143859863
petismo	0.8173528909683228

Lastly, we'll present the data relating words categorized by grammatical classification and their coefficient relation to certain keywords.

Verbs closer to Haddad:

Word	Coefficient
vestir	0.9855930805206299
iniciou	0.9852603673934937
desabar	0.974722146987915
nasceram	0.9517772197723389
divulgou	0.9479817152023315
especula	0.9463067650794983
expedir	0.9430577754974365
propaga	0.9317755699157715
identificado	0.9297734498977661
criminaliza	0.9228401184082031
leio	0.9228401184082031

Verbs closer to Bolsonaro:

Word	Coefficient
acompanhado	0.9687446355819702
excluir	0.9616330862045288
representa	0.9570780396461487
acionar	0.9462116360664368
pacifica	0.9434142112731934
errar	0.9410009980201721
permitir	0.9223222732543945
conferir	0.9179933667182922
criticou	0.9092496037483215
frear	0.8937103152275085
enquadrado	0.8924423456192017

Adjectives closer to Haddad:

Word	Coefficient
ruralista	0.947772324085235
guerrilheiro	0.939964234828949
nulo	0.939508855342865
oficiais	0.9365040063858032
promissor	0.9309912919998169
relevante	0.9306001663208008
breve	0.9161219000816345
duplo	0.906759738922119
devido	0.8841083645820618
imperial	0.8825623393058777
atuante	0.8924423456192017

Adjectives closer to Bolsonaro:

Word	Coefficient
presencial	0.9803654551506042
preciso	0.9751306772232056
americanas	0.9637729525566101
conhecida	0.9259804487228394
preocupantes	0.909659385681152
referido	0.9306001663208008
longa	0.9041167497634888
nazista	0.8900717496871948
brasileira	0.8846299648284912
respectivo	0.881061315536499
empresarial	0.8795811533927917
religioso	0.8624833822250366 height

Capítulo 7

Conclusion

7.1 Inquiring about results found

In this paper, we've introduced the background on why the discussion about the role of search engines and technology companies for that matter is a key one for the shaping of democratic societies in the 21st century. We laid out a framework through which we were able to monitor the Brazilian general election of 2018, building accounts for different categories (ideological, neutral and gender) and then establishing a training procedure which aimed to make Google used to the way through which each account behaved. From this effort we built a dataset consisting of about three hundred thousand observations, with a hundred thousand being considered for our analysis.

As a next step, we applied the Word2Vec algorithm to each news' titles in our dataset, separating between left-wing and right-wing accounts, which led to a few initial insights, such as observing 'communist' as the most related term to a left-wing candidate due to the campaign efforts of his opponent, whose most related term is 'presidenciavel', which refers to someone who is fit to being elected as president.

Although this effort produced valid results, we proceeded to build a script which visited each URL in our dataset, downloaded and saved their textual contents to a CSV file which was later separated in three text-based files with which we trained another instance of Word2Vec. This new set of data consisted of a general file, containing all text we gathered from each page regardless of profile categories. Then we created two new datasets which consisted of text related to right-wing accounts and left-wing accounts. Then we proceeded to build Word2Vec models of each of them and observe terms most related to the two presidential candidates who ran for office and a few terms which were central to the debate. Overall we consider the work presented here to provide an initial look inside one of the most controversial Brazilian elections ever. Due to the widespread use of technology unavailable in previous elections such as messaging apps, as well as digital phenomena which didn't play such a key role as in this one as *fake news*, to fully understand how this election unfolded and the influence of digital services in our democracy such as Google Search would ideally require years of effort with datasets containing billions of records. Since there isn't such database yet available, we believe ours is the first to thoroughly analyze and record the interaction between users, even if not real ones, and search engines in the context of a real life, 100M voters election. Moreover, the analysis we provided helps us see the bigger picture of what that election presented as unique from other voting processes: The highly polarized voter behavior, which clearly reflected in their online lives and consequently their Google searches. Moreover, we performed such an analysis with an algorithm which reportedly works better with much larger datasets, so to be able to explore an intrincate setting as a major election using it the main engine also seems to be an interesting discovery.

Finally, we began this research effort in the quest to answer whether or not different results were shown to users belonging to different ideological categories, to which we believe the answer is yes. Although it's widely known Google adapts its results to users with different political profiles even when they search the same query might lead to results that impact the election for the better or the worse. Therefore, we believe this is the first study to raise these questions for Brazilian elections and we've done our best to answer the questions proposed with the resources available, which goes from man hours to computing power and dataset sizes. We hope this is the first of many explorations which will help define the relationship between democracies and digital tools as a beneficial one to both companies that run such services and more importantly to the hundreds of thousands of citizens who rely on such mechanisms to keep informed and decide their votes and political opinions.

7.2 Future work

As we see it, there are at least two possible paths for future research: The first one being to enhance the amount of data available to run Word2Vec against. As said before, the smallest database to which Word2Vec is run against in the original paper contains 24 million words, while our database contains approximately 3 million records. We believe that with a larger body of text, relationships between important words would become clearer than we were able to see in this study. Secondly, one can probably find interesting results by applying other algorithms rather than Word2Vec to either our database or a larger one and compare the results, providing in last instance a more robust analysis for the scientific community and society as a whole.

Referências

[1] Morgan, L Lloyd, Santosh Kesari e Devra Lee Davis: *Why children absorb more microwave radiation than adults: The consequences.* Journal of Microscopy and Ultrastructure, 2(4):197–204, 2014. 2

[2] Strasburger, Victor C, Amy B Jordan e Ed Donnerstein: *Health effects of media on children and adolescents.* Pediatrics, 125(4):756–767, 2010. 2

[3] Buolamwini, Joy e Timnit Gebru: *Gender shades: Intersectional accuracy disparities in commercial gender classification.* Em *Conference on Fairness, Accountability and Transparency*, páginas 77–91, 2018. 4

[4] 4

[5] Times, The New York: *The plot to subvert an election,unraveling the russia story so far.* `https://www.nytimes.com/interactive/2018/09/20/us/politics/russia-interference-election-trump-clinton.html`, acesso em 2019-01-27. 4

[6] Brin, Sergey e Lawrence Page: *The anatomy of a large-scale hypertextual web search engine.* Computer networks and ISDN systems, 30(1-7):107–117, 1998. 7

[7] Sartor, Giovanni e Mario Viola de Azevedo Cunha: *The italian google-case: Privacy, freedom of speech and responsibility of providers for user-generated contents.* International Journal of Law and Information Technology, 18(4):356–378, 2010. 9

[8] Bond, Robert M, Christopher J Fariss, Jason J Jones, Adam DI Kramer, Cameron Marlow, Jaime E Settle e James H Fowler: *A 61-million-person experiment in social influence and political mobilization.* Nature, 489(7415):295, 2012. 10

[9] Gayo-Avello, Daniel: *A meta-analysis of state-of-the-art electoral prediction from twitter data.* Social Science Computer Review, 31(6):649–679, 2013. 10

[10] Metaxas, Panagiotis T, Eni Mustafaraj e Dani Gayo-Avello: *How (not) to predict elections.* Em *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*, páginas 165–171. IEEE, 2011. 10

[11] Trevisan, Filippo, Andrew Hoskins, Sarah Oates e Dounia Mahlouly: *The google voter: search engines and elections in the new media ecology.* Information, Communication & Society, 21(1):111–128, 2018. 10

[12] Lui, Catherine, P Takis Metaxas e Eni Mustafaraj: *On the predictability of the us elections through search volume activity*. 2011. 11

[13] Epstein, Robert e Ronald E Robertson: *The search engine manipulation effect (seme) and its possible impact on the outcomes of elections*. Proceedings of the National Academy of Sciences, 112(33):E4512–E4521, 2015. 11

[14] Xue, Bai, Chen Fu e Zhan Shaobin: *A study on sentiment computing and classification of sina weibo with word2vec*. Em *2014 IEEE International Congress on Big Data*, páginas 358–363. IEEE, 2014. 11

[15] Skymind.ai: *A beginner's guide to neural networks and deep learning*. `https://skymind.ai/wiki/neural-network`, acesso em 2019-04-03. 18, 19

[16] Mikolov, Tomas, Kai Chen, Greg Corrado e Jeffrey Dean: *Efficient estimation of word representations in vector space*. arXiv preprint arXiv:1301.3781, 2013. 21, 23, 43

[17] Pan, Bing, Helene Hembrooke, Thorsten Joachims, Lori Lorigo, Geri Gay e Laura Granka: *In google we trust: Users' decisions on rank, position, and relevance*. Journal of computer-mediated communication, 12(3):801–823, 2007. 27

[18] Linden, John e Tobias Teeter: *Method for performing real-time click fraud detection, prevention and reporting for online advertising*, 2012. US Patent 8,321,269. 30

[19] Brindley, Richard e Toby Doig: *Fraud prevention and detection for online advertising*, 2014. US Patent 8,719,396. 30

Apêndice A

Apêndice 1

A.0.1 Full list of stopwords

de a o que e do da em um para com não uma os no se na por mais as dos como mas ao ele das à seu sua ou quando muito nos já eu também só pelo pela até isso ela entre depois sem mesmo aos seus quem nas me esse eles você essa num nem suas meu às minha numa pelos elas qual nós lhe deles essas esses pelas este dele tu te vocês vos lhes meus minhas teu tua teus tuas nosso nossa nossos nossas dela delas esta estes estas aquele aquela aqueles aquelas isto aquilo estou está estamos estão estive esteve estivemos estiveram estava estávamos estavam estivera estivéramos esteja estejamos estejam estivesse estivéssemos estivessem estiver estivermos estiverem hei há havemos hão houve houvemos houveram houvera houvéramos haja hajamos hajam houvesse houvéssemos houvessem houver houvermos houverem houverei houverá houveremos houverão houveria houveríamos houveriam sou somos são era éramos eram fui foi fomos foram fora fôramos seja sejamos sejam fosse fôssemos fossem for formos forem serei será seremos serão seria seríamos seriam tenho tem temos tém tinha tínhamos tinham tive teve tivemos tiveram tivera tivéramos tenha tenhamos tenham tivesse tivéssemos tivessem tiver tivermos tiverem terei terá teremos terão teria teríamos teriam embora yt lin famososcade afa jaworski ce asked anuncie fraca teus won alto l clicar gt bichos been newsletterem main cantando street okngroup tap quilometro bone km kabum content disabled login oamento lost h k car Gary most povocopyright the obs igshid ador et aguiar along one want jun growth institucionalatendimentoconexao due