UNIVERSIDADE FEDERAL DO CEARÁ

INSTITUTO UFC VIRTUAL

CURSO DE GRADUAÇÃO EM SISTEMAS E MÍDIAS DIGITAIS

MATHEUS OLIVEIRA COSTA

DESCOBERTA DE PERFIS DE *YOUTUBERS* VIA CLUSTERIZAÇÃO

FORTALEZA

2019

MATHEUS OLIVEIRA COSTA

DESCOBERTA DE PERFIS DE *YOUTUBERS* VIA CLUSTERIZAÇÃO

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistemas e Mídias Digitais da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Sistemas e Mídias Digitais.

Orientadora: Profa. Dra. Ticiana Linhares Coelho da Silva

FORTALEZA

2019

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

C874d Costa, Matheus Oliveira.
Descoberta de Perfis de Youtubers via Clusterização / Matheus Oliveira Costa. – 2019.
45 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Instituto UFC Virtual, Curso de Sistemas e Mídias Digitais, Fortaleza, 2019.
Orientação: Profa. Dra. Ticiana Linhares Coelho da Silva.

1. Clusterização. 2. YouTube. 3. Processamento de Linguagem Natural. 4. Word Embeddings. I. Título.
CDD 302.23

MATHEUS OLIVEIRA COSTA

DESCOBERTA DE PERFIS DE *YOUTUBERS* VIA CLUSTERIZAÇÃO

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistemas e Mídias Digitais da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Sistemas e Mídias Digitais.

Aprovada em:

BANCA EXAMINADORA

Profa. Dra. Ticiana Linhares Coelho da Silva (Orientadora)
Universidade Federal do Ceará (UFC)

Profa. Dra. Georgia da Cruz Pereira
Universidade Federal do Ceará (UFC)

Prof. Dr. Leonardo Oliveira Moreira
Universidade Federal do Ceará (UFC)

AGRADECIMENTOS

À Deus antes de tudo.

À minha família, que me apoia e acredita no meu potencial em todas as situações e dificuldades. Em especial a minha amada irmã Isabela, minha companheira de todos os dias, sempre escutando minhas besteiras e fazendo o possível para criar os momentos mais felizes e engraçados.

À Professora Ticiana por todo o período de orientação e instrução para concretização deste trabalho. E aos professores da banca, Geórgia e Leonardo, que tive o prazer de participar de cadeiras ministradas, concederam seu tempo para avaliar e contribuir com este trabalho.

Aos meus amigos e colegas de curso e de grupos extracurriculares, em especial ao Mário, Matheus Campelo, João Paulo, William, André Albuquerque, Alexandre e Carol, que estiveram mais presentes e contribuíram durante essa jornada, cada um de sua maneira e são amizades que levo da universidade para a vida.

À Universidade Federal do Ceará, por todos os anos de educação de qualidade, me proporcionando crescer tanto profissionalmente, quanto como pessoa.

“As coisas belas são difíceis”

(Platão)

RESUMO

Aplicações na Internet como o YouTube permitem os indivíduos criarem conteúdo e colocarem na rede para outros verem e interagirem de forma praticamente livre e fácil. Nos últimos anos tem sido produzido um grande volume de informações geradas pelos seus usuários, que com frequência compartilham seus sentimentos, opiniões e acontecimentos em seus vídeos. Isso possibilita o desenvolvimento de aplicações que usam dessas informações em outros tipos de sistemas de recomendação que se baseiam no conteúdo de suas sugestões, e ainda, auxiliar no entendimento de como a comunicação das pessoas está mudando com o passar dos anos. No entanto, analisar esse grande volume de dados de forma não automatizada consiste em um problema não trivial. Seguindo esta motivação, este trabalho propõe utilizar técnicas e algoritmos de Aprendizagem de Máquina para descobrir e identificar os perfis de *youtubers*, mas especificamente a clusterização. Para isso, são utilizados os textos das legendas dos vídeos geradas automaticamente pelo YouTube como base dos dados, e depois agrupados os conteúdos abordados baseada em uma função de similaridade. A partir dos resultados, foi descoberto que é possível utilizar técnicas de clusterização para encontrar grupos de *youtubers* similares e em combinações desconhecidas anteriormente. Ao final da pesquisa, foi concluído que é possível utilizar técnicas simples de clusterização para encontrar grupos de *youtubers* se baseando no conteúdo de seus vídeos de uma forma minimamente satisfatória.

Palavras-chave: Clusterização. YouTube. Processamento de Linguagem Natural. *Word Embeddings*

ABSTRACT

Internet applications such as YouTube allow individuals to create content and post to others for viewing and interacting virtually freely and easily. Over the past few years, a wealth of user-generated information has been produced that often shares their feelings, opinions, and events in their videos. This enables the development of applications that use this information in other types of recommendation systems that are based on the content of your suggestions, and also assist in the understanding of how communication people are changing over the years. However, analyzing this large volume of data unauthorized is a nontrivial problem. Following this motivation, this paper proposes to use Machine Learning techniques and algorithms to discover and identify youtuber profiles, but specifically clustering. To do this, the video subtitles texts generated automatically by YouTube are used as the database, and then the content covered is grouped based on a similarity function. From the results, it was found that it is possible to use clustering techniques to find groups of similar youtubers in previously unknown combinations. At the end of the research, it was concluded that simple clustering techniques can be used to find groups of *youtubers* based on the content of your videos in a minimally satisfactory way.

Keywords: Clustering. YouTube. Natural Language Processing. Word Embeddings.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

API	*Application Programming Interface*
GloVe	*Global Vectors for Word Representation*
GRIM	Grupo de Pesquisa da Relação Infância, Adolescência e Mídia - UFC
HTTP	*Hypertext Transfer Protocol*
IA	Inteligência Artificial
IDF	*Inverse Document Frequency*
JSON	*JavaScript Object Notation*
NILC	Núcleo Interinstitucional de Linguística Computacional da USP
TF	*Term Frequency*
TF-IDF	*Term Frequency-Inverse Document Frequency*
URL	*Uniform Resource Locator*

SUMÁRIO

1 INTRODUÇÃO

O advento da Web 2.0 evidenciou uma transformação nas formas de relacionamento e interação da sociedade *online*. As novas aplicações permitem os indivíduos criarem conteúdos e colocarem na rede para outros verem e interagirem de forma praticamente livre e fácil. Um dos maiores representantes dessa nova realidade, é o *site* de compartilhamento de vídeos YouTube.

Criado em 2005, o YouTube permite que qualquer pessoa possa enviar, assistir e interagir com outros usuários na sua rede, de forma gratuita e simples. O serviço atingiu um enorme sucesso logo nos seus primeiros anos de existência (REUTERS, 2006) e por conta disso, em 2007 a plataforma foi adquirida pelo Google (JOHNSON; SWENEY, 2006). Atualmente, o YouTube engloba 95% da população total da internet (YOUTUBE, 2019) e é o segundo *site* mais acessado do mundo (ALEXA, 2019).

O diferencial da plataforma foi possibilitar que qualquer um pudesse enviar, assistir e compartilhar vídeos *online*. Ao facilitar essa etapa da produção somado ao fato de pouco intervir nos assuntos abordados nos vídeos, possibilitou milhares de pessoas a criar e disponibilizar conteúdo original *online*. O YouTube também conta com funcionalidades limitadas de rede social para promover conexões entre os usuários. Apesar de ser um diferencial, não é algo suficiente para caracterizá-lo como ambiente de interação *online*, o que não foi um limitante para os criadores de conteúdo pensarem em criar formas diferenciadas de interagir com a comunidade (CHAU, 2010).

Portanto, é possível observar que o sucesso do YouTube também depende muito de seus criadores de conteúdo, os também chamados de *youtubers*[1]. Por conta dessas características da rede social, a conexão entre os criadores para a plataforma e seus inscritos é muito próxima e transparente, lembrando até algo pessoal.

1.1 Problema e Problemática

O YouTube também conta com um complexo sistema de recomendação, que se baseia no perfil de uso dos usuários para indicar vídeos relacionados (COVINGTON *et al.*, 2016). Além disso, possui a recomendação de canais de outros *youtubers* similares, no entanto, somente está visível na página do canal caso o responsável tenha ativado. Essa classificação também se baseia nos metadados dos vídeos do criador em específico (SIMONET, 2013), ou seja, ignora o

[1] Eles são os protagonistas dos vídeos e que alcançaram a fama por meio da plataforma, muitos deles sem nenhuma experiência e desconhecidos anteriormente.

real conteúdo do vídeo.

Por ser uma das maiores ferramentas existentes de compartilhamento de conteúdo, compreender o perfil dos criadores de conteúdo, os *youtubers* também pode auxiliar em como a comunicação falada está mudando com o passar dos anos. Questões como essas já são objetos de estudos pelo Grupo de Pesquisa da Relação Infância, Adolescência e Mídia - UFC (GRIM) em seus projetos de pesquisa e o YouTube não provê ferramentas suficientes para auxiliar na coleta e processamento dos dados necessários e de forma satisfatória para as pesquisas. Portanto, são necessárias soluções que trabalhem no problema descrito, bem como possa ser utilizada em colaboração com esses tipos de pesquisa.

Previamente ao início do trabalho foi realizada uma busca por pesquisas anteriores ou projetos que atacam o problema descrito, esta etapa é aprofundada posteriormente neste trabalho no Capítulo 3. No entanto, não foram encontradas soluções ou ferramentas que executem a ação de relacionar vídeos aos seus criadores de forma a classificar e agrupar tais vídeos de acordo com o perfil do *youtuber*. Isso pode ser relevante para sistemas de recomendação de canais, *youtubers* e aos usuários da plataforma. Como também, algo de tal natureza é interessante para profissionais da área de comunicação. Os termos e expressões utilizadas em um texto, como também sua conjuntura linguística são importantes para compreender a linguagem. Por isso, esse processo de agregação textual baseado na similaridade em etapas preliminares dos estudos e análises dos profissionais, pode promover análises mais cuidadosas.

Este trabalho visa propor uma solução por meio da análise dos textos das legendas dos vídeos do YouTube, de tal sorte que canais ou *youtubers* de alta similaridade (ou mesmo perfil) estejam em um mesmo grupo. Bem como, canais ou *youtubers* dissimilares estejam em grupos diferentes.

1.2 Justificativa

Clusterização de Dados refere-se a um conjunto de técnicas de Mineração de Dados (em inglês, *Data Mining*) que pode ser definida como o processo de dividir grandes conjuntos de dados em subconjuntos chamados *clusters*. A divisão é feita baseada no algorítimo de análise escolhido, de forma que os objetos dentro de um *cluster* são suficientemente semelhantes entre si e diferentes dos objetos nos outros *clusters*. Nesse contexto, diferentes métodos de análise podem gerar diferentes agrupamentos no mesmo conjunto de dados. Consequentemente, o processo de análise é útil, pois pode levar a descoberta de grupos de dados anteriormente desconhecidos

(HAN *et al.*, 2011).

As técnicas de Aprendizagem de Máquina são usadas em muitas aplicações em diferentes contextos, inclusive análise de conteúdo *online*. Trabalhos anteriores demonstram a utilização dos métodos para descobrir perfis extremistas em *blogs* (CHAU; XU, 2007) e vídeos (SUREKA *et al.*, 2010), para o reconhecimento de padrões em vídeos baseados no textos derivados deles (CHANG *et al.*, 2005) e também para agregar perfis de uso para uma experiência personalizada (MOBASHER *et al.*, 2002).

1.3 Objetivos

O objetivo geral desse trabalho consiste em agrupar perfis de canais ou *youtubers* baseado na análise das falas dos vídeos utilizando técnicas de clusterização. Dessa forma, descobrir grupos de canais ou *youtubers* que apresentem alta similaridade entre si. Adicionalmente, tem como objetivos específicos a coleta e disponibilização de uma base de textos com as legendas de diferentes vídeos do YouTube.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo apresentar os principais temas abordados neste trabalho. Em cada um dos subcapítulos posteriores são definidos os conceitos necessários ao entendimento do assunto abordado e consequentemente aplicados para a realização da solução proposta.

2.1 Aprendizagem de Máquina

Aprendizagem de Máquina (ou *Machine Learning*, em inglês) evoluiu como um subcampo de estudo da Inteligência Artificial (IA) que trabalha com algoritmos de autoaprendizagem que usam o conhecimento derivado de dados para fazer predições. Em vez de exigir que os humanos produzam regras manualmente, técnicas de aprendizagem de máquina oferecem uma alternativa mais eficiente para descobrir conhecimento em grandes volumes de dados, e ainda melhorar gradualmente o desempenho de modelos preditivos e de tomada de decisões (RASCHKA; MIRJALILI, 2017).

Os modelos de aprendizagem de máquina podem ser divididos em três tipos: aprendizagem supervisionada, aprendizagem não-supervisionada e aprendizagem por reforço (RASCHKA; MIRJALILI, 2017).

No **aprendizado supervisionado**, o modelo se baseia em aprender de um conjunto de dados de treinamento rotulados e com resultados previamente definidos, para dessa forma, identificar padrões que possibilitem fazer predições de informações desconhecidas ou futuras. No processo de aprendizado, existem duas subcategorias: classificação e regressão. Na *classificação* tem como objetivo prevê as classificações de novas instâncias, baseadas em observações passadas. Enquanto na *regressão* acontece a predição em resultados contínuos. Nesse caso, é dado um número de variáveis explicativas e uma variável contínua. Depois é buscado encontrar uma relação entre as variáveis, permitindo prever o resultado (RASCHKA; MIRJALILI, 2017).

Na **aprendizagem por reforço**, o objetivo é desenvolver um sistema (*agente*) que aprimora sua performance baseada em interações com o *ambiente*. Por meio da interação com o ambiente, um agente pode usar da técnica para aprender uma série de ações que maximizam uma função que calcula a recompensa, por meio de um abordagem de tentativa e erro exploratória ou planejamento deliberativo (RASCHKA; MIRJALILI, 2017).

Por fim, existe o **aprendizado não-supervisionado** em que os dados são não-

rotulados ou sem uma estrutura definida. Ao usar técnicas desse tipo, é possível explorar a estrutura dos dados para extrair informações significativas sem um resultado conhecido ou uma função de recompensa (RASCHKA; MIRJALILI, 2017).

2.2 Clusterização de Dados

Clusterização é um processo de agrupar um grande conjunto de objetos em múltiplos grupos menores ou *clusters* de modo que objetos dentro de um *cluster* tenham alta similaridade, mas sejam muito diferentes dos objetos em outros *clusters*. Dissimilaridades e similaridades são avaliadas com base nos valores de atributos que descrevem os objetos e frequentemente envolvem medidas de distância (HAN *et al.*, 2011). No entanto, clusterização é definida como uma classificação não-supervisionada de dados, pois as informações presentes não estão rotuladas e não existe conhecimento prévio dos membros dos grupos (RASCHKA; MIRJALILI, 2017).

As medidas de similaridade e dissimilaridade são referidas como medidas de proximidade e ambas estão relacionadas. Uma medida de similaridade para dois objetos, i e j, normalmente retornará um valor próximo de zero se os objetos forem dissimilares. Quanto maior o valor de similaridade, maior a similaridade entre os objetos (normalmente, o valor 1 indica semelhança completa, ou seja, os objetos são idênticos) (HAN *et al.*, 2011).

Como um *cluster* é uma coleção de objetos de dados que são semelhantes entre si e diferentes de objetos em outros *clusters*, um *cluster* pode ser tratado como uma classe implícita. Nesse sentido, a clusterização é às vezes chamada de classificação automática. A diferença entre estas técnicas é que a clusterização encontra grupos (ou classes) automaticamente. A clusterização também é intitulada de segmentação de dados em algumas aplicações, por conta do particionamento de grandes conjuntos de dados em grupos ser baseado de acordo com a similaridade (HAN *et al.*, 2011).

Como um método de mineração de dados, a clusterização pode ser usada como uma ferramenta autônoma para obter informações sobre a distribuição de dados, observar as características de cada *cluster* e focar em um determinado conjunto de *clusters* para análise posterior. Como alternativa, ela pode servir como uma etapa de pré-processamento para outros algoritmos, como os de classificação. A clusterização também pode ser usada para detecção de *outliers* (valores distantes de qualquer *cluster*) em contextos que eles podem ser mais interessantes do que os casos comuns (HAN *et al.*, 2011).

Na mineração de dados, os esforços concentraram-se em encontrar métodos para

clusterização eficientes e eficazes para grandes volumes de dados. Temas ativos de pesquisa enfocam a escalabilidade de métodos de clusterização, a eficácia de métodos para agrupar formas complexas e tipos de dados (por exemplo, texto, gráficos e imagens), técnicas de clusterização para dados de alta dimensionalidade, entre outros problemas (HAN *et al.*, 2011).

2.2.1 K-means

K-means é um algoritmo simples de implementar e ao mesmo tempo computacionalmente muito eficiente comparado a outros algoritmos de clusterização, o que justifica sua popularidade. Ele pertence a categoria de Clusterização Baseada em Protótipo (*Prototype-based Clustering*), que significa que cada *cluster* é representado por um protótipo, que pode ser tanto um centróide (média) de pontos similares, ou um *medoid* (ponto mais frequente ou representativo) (RASCHKA; MIRJALILI, 2017).

O funcionamento do algoritmo pode ser resumido em quatro passos:

1. Escolher aleatoriamente *cluster* centróides nos pontos da amostra como centros iniciais dos *clusters*;
2. Atribuir cada ponto da amostra ao seu centróide mais próximo;
3. Mover os centróides ou medóides para o centro dos pontos que foram atribuídos a ele;
4. Repetir os passos 2 e 3 até que as atribuições dos *clusters* não sejam alteradas ou uma tolerância definida ou número máximo de iterações definidas sejam atingidas (RASCHKA; MIRJALILI, 2017).

É possível definir similaridade como o oposto da distância, e uma função de distância frequentemente utilizada em clusterização é a Distância Euclidiana. Esta distancia é computada pelo soma do valor quadrático da diferença em cada dimensão de dois pontos x e y em um espaço m-dimensional. Baseada nessa medida de distância, é possível descrever o *k-means* como um problema de otimização simples, uma vez que iterativamente tenta-se minimizar a Soma dos Erros Quadráticos, também referida como a inércia de um *cluster*, como demonstrado na Equação 2.1.

$$SSE = \sum_{i=1}^{n} \sum_{j=1}^{k} w^{(i,j)} \left\| x^{(i)} - \mu^{(j)} \right\|_2^2 \tag{2.1}$$

Nesse caso, $\mu^{(j)}$ é o centróide para o *cluster* j, assim, $w^{(i,j)} = 1$ se a amostra $x^{(i)}$

está no *cluster* j e $w^{(i,j)} = 0$ caso contrário (RASCHKA; MIRJALILI, 2017).

Enquanto o *k-means* é muito bom em identificar *clusters* de forma esférica, uma das desvantagens desse algoritmo de agrupamento é que temos que especificar o número de *clusters*, ou seja *k*, *a priori*. Uma escolha inadequada para *k* pode resultar em um mau desempenho na clusterização. O número de *clusters* a escolher nem sempre pode ser tão óbvio em aplicações do mundo real, especialmente ao trabalhar com um conjunto de dados dimensionalmente mais alto e que não pode ser visualizado. As outras propriedades do *k-means* são, primeiramente, que os *clusters* não se sobrepõem, nem são hierárquicos, e que há pelo menos um item em cada *cluster* (RASCHKA; MIRJALILI, 2017).

2.3 *Word Embedding*

Word Embedding é definido como um conjunto de técnicas para geração de representações vetoriais de palavras, derivadas de vários métodos de treinamento inspirados em modelos neurais de linguagem (LEVY; GOLDBERG, 2014). As representações oriundas desse modelo conseguem codificar regularidades sintáticas e semânticas com alta precisão e preservam as regularidades lineares entre as palavras (MIKOLOV *et al.*, 2013a).

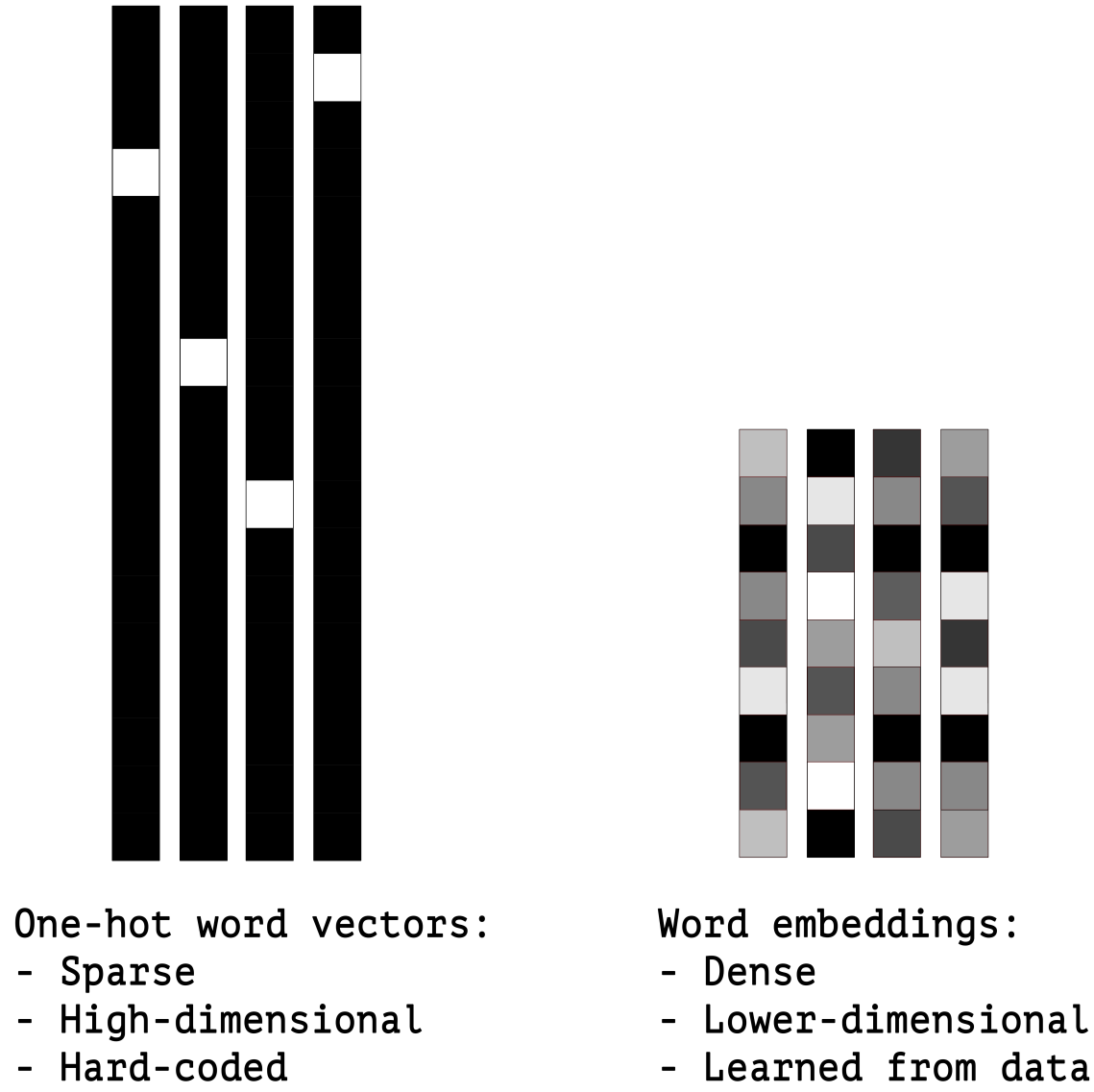
O modelo vem em contraposição a outras abordagens como a *One-hot encoding*, em que são armazenados valores binários em vetores de grandes dimensões (mesmas dimensões que o número de palavras do vocabulário). Já os *Word Embedding* são vetores de pontos flutuantes e consequentemente com menor quantidade de valores (as abordagens comumente usadas variam entre 50-100 elementos). Dessa forma, é criado um vetor de densidade para cada palavra, de tal sorte que palavras similares (ou que aparecem em contextos similares) têm representação vetorial similar (CHOLLET, 2017).

Na Figura 1 é demonstrado um comparativo entre representações obtidas pelas técnicas *one-hot encoding* e *word embedding*. Enquanto na primeira são representações esparsas e de grandes dimensões, *word embeddings* são densas, de menores dimensões e geradas por aprendizagem a partir dos dados.

A ideia de um espaço de incorporação denso e de baixa dimensão para palavras, computado de uma maneira não supervisionada, foi inicialmente explorada em Bengio *et al.* (2003), mas apenas começou a aparecer em aplicações de pesquisa e industria após o desenvolvimento do algoritmo *Word2vec*[1], demonstrado em Mikolov *et al.* (2013a) e Mikolov *et*

[1] Disponível em <https://code.google.com/p/word2vec/>. Acesso em: 27 de Dez. de 2019.

Figura 1 – Figura com o comparativo entre representações obtidas por *one-hot encoding* e *word embedding*.



Fonte: (ALLAIRE, 2018)

al. (2013b) (CHOLLET, 2017).

A seguir, é fornecido mais detalhes sobre o GloVe, que tem sido amplamente utilizada para geração de *word embeddings*.

2.3.1 GloVe

Global Vectors for Word Representation (GloVe) é um algoritmo de aprendizagem não supervisionado para obter representações vetoriais de palavras. Ele usa um modelo de regressão log-bilinear global para a obtenção das representações e, por isso, supera outros

modelos nas tarefas de analogia, semelhança de palavras e tarefas de reconhecimento de entidades nomeadas. Para o mesmo corpus, vocabulário, tamanho de janela e tempo de treinamento, o GloVe supera consistentemente o *Word2vec*, pois é mais rápido e também obtém os melhores resultados (PENNINGTON *et al.*, 2014).

O algoritmo foi disponibilizado em código aberto por pesquisadores da Universidade de Stanford (os desenvolvedores originais) e dessa forma possibilitou ser usado para criar representações de palavras em diversos vocabulários ao redor do mundo. Dentre essas, existe a representação disponibilizada pelo Núcleo Interinstitucional de Linguística Computacional da USP (NILC) para a língua portuguesa, onde foram avaliados diferentes modelos de incorporação de palavras treinados em um grande corpus português, incluindo variantes brasileiras e europeias. Foram geradas representações utilizando quatro técnicas de *Word Embedding*[2], dentre elas o GloVe (HARTMANN *et al.*, 2017).

2.4 TF-IDF

A chamada TF-IDF é uma medida estatística utilizada quando se tem como alvo descobrir se uma palavra contém uma informação discriminatória ou útil dentro de um corpus de documentos. Ela definida como o produto da Frequência do Termo (*Term Frequency* (TF)) e a Frequência Inversa no Documento (*Inverse Document Frequency* (IDF)) (RASCHKA; MIRJALILI, 2017). Existem algumas diferenças pequenas no procedimento formal para implementar o TF-IDF em torno de suas aplicações, mas sua proposta geral pode ser demostrada na Equação 2.2.

$$tf - idf(t,d) = tf(t,d) \times idf(t,d) \tag{2.2}$$

A frequência do termo $tf(t,d)$ significa a quantidade de vezes que o termo t acontece em um documento d. Enquanto a frequência inversa no documento $idf(t,d)$ é quantidade de documentos (n_d) dividido pelo número de documentos d que contém o termo t (RASCHKA; MIRJALILI, 2017). A Equação 2.3 demostra como o $idf(t,d)$ pode ser calculado.

$$idf(t,d) = \log \frac{n_d}{1+df(d,t)} \tag{2.3}$$

Nesse caso, n_d é o número total de documentos que compõem o corpus linguístico para a análise, e $df(d,t)$ é a quantidade de documentos d que contem o termo t. É frisado que

[2] <http://nilc.icmc.usp.br/embeddings>. Acesso em: 27 de Dez. de 2019.

a adição a constante 1 ao denominador é opcional e tem apenas como propósito de atribuir um valor diferente de zero aos termos que ocorrem em todas as amostras do treinamento. Por último, a função logarítmica é empregada para garantir que não seja dado muito peso as baixas frequências nos documentos (RASCHKA; MIRJALILI, 2017).

Apesar de ser uma métrica confiável, existem algumas situações diferentes que podem ocorrer para cada palavra, dependendo dos valores do $tf(t,d)$, n_d e $df(d,f)$. Uma dessas situações é quando o tamanho do corpus é aproximadamente igual a frequência de uma palavra nesse mesmo corpus. Se $1 < \log \frac{n_d}{1+df(d,f)} < c$ para uma pequena constante c, então o TF-IDF será menor que o $tf(t,d)$, como também ainda será positivo. Isso implicará que a palavra é relativamente comum no corpus, no entanto, ainda tem considerável importância no mesmo (RAMOS *et al.*, 2003).

Outro caso é quando o $tf(t,d)$ é alto e o $df(d,f)$ baixo. Logo o $\log \frac{n_d}{1+df(d,f)}$ resultará em um valor alto, ocasionando um TF-IDF igualmente grande. Ao contrário do outro, esse é um caso que interessa pois já que palavras com altos métricas de TF-IDF significam que são importantes em documentos específicos e pouco comuns no corpus de documentos (RAMOS *et al.*, 2003).

3 TRABALHOS RELACIONADOS

Existem alguns trabalhos, em sua grande maioria na língua inglesa, que se assemelham em algum grau com a área e objetivos deste trabalho. Assim sendo, neste capítulo são enumerados alguns desses trabalhos e comparado alguns métodos aplicados neles com os aplicados neste trabalho.

Todos os trabalhos foram elencados após uma busca em repositórios *online* utilizando palavras-chave ou termos como "YouTube", "clustering", "*clustering YouTube videos*", "*grouping youtubers*", "*find similar youtubers*" e "*word embedding and clustering*". Portanto são artigos procuram de alguma forma identificar, descobrir padrões ou entender relacionamentos entre os dados. E alguns utilizam o YouTube como fonte dos dados para as análises.

Chau e Xu (2007) propõem uma abordagem semiautomática para estudar os relacionamentos entre grupos de ódio na Internet. A proposta é composta por módulos onde primeiramente é utilizado uma ferramenta para a extração das informações em uma rede de *blogs*. Os dados extraídos e seus relacionamentos são passados por um processo de análise posterior. Por fim, há uma etapa de visualização dos resultados na análise anterior. O processo é composto por uma análise topológica (para garantir que a rede extraída é significativa para as outras análises), uma análise para identificar os pontos chave da rede e, por fim, uma análise na comunidade para identificar grupos sociais na rede. Os resultados do processo apresentado se mostraram satisfatórios e o mesmo pode ser utilizado para auxiliar no monitoramento de tais atividades, bem como trouxe esclarecimentos sobre as propriedades estruturais dos grupos de ódio *online* e ajudou a aprofundar a compreensão sobre tal movimento.

Sureka *et al.* (2010) apresentam um conjunto de métodos para analisar o acervo do YouTube para detectar vídeos, usuários e comunidades de promoção do ódio. No trabalho, são usados para a análise as propriedades dos vídeos (número de visualizações, comentários, avaliações positivas e negativas, duração etc.), conteúdo dos comentários nos vídeos de forma a descobrir os termos mais utilizados e relacionamentos na rede social da plataforma (inscrições, amizades, vídeos favoritados e *playlists* criadas). Com a solução proposta é possível de descobrir usuários centrais e influentes, vídeos e comunidades escondidas usando as técnicas de análise na rede social.

Chaudhary e Sureka (2013) demonstram um classificador que detecta *spams*[1] automaticamente nas vídeo-respostas de um vídeo do YouTube. O classificador trabalha com vídeos

[1] Conteúdo irrelevante ou inapropriado enviado na Internet para um grande número de recipientes.

nos contextos de reconhecimento de vídeos promocionais, pornográficos e vídeos enviados por *scripts* automatizados ou *botnet*[2]. A análise desenvolvida no trabalho revelou que certas características linguísticas (presença de termos no título ou descrição), temporais (duração e momento de envio dos vídeos) e baseadas em popularidade (número de inscritos, reações e visualizações) podem ser usadas para prever o tipo do vídeo. Essas características serviram de base para o classificador reconhecer os vídeos de *spam*. A proposta apresentada conseguiu obter resultados com 80% de precisão em um conjunto de dados experimental, provando que a presença de indicadores nos metadados do vídeo podem servir para reconhecer automaticamente *spam* em um vídeo-resposta.

Aggarwal *et al.* (2014) apresentam uma abordagem para identificar vídeos de assédio, invasão de privacidade e contravenção no YouTube, por meio da mineração de seus metadados. Os pesquisadores conduziram um estudo de caracterização que manualmente caracterizou vídeos de um conjunto de dados amostrais obtidos por meio da *Application Programming Interface* (API) do YouTube. Foram definidas diversas características discriminatórias que abrangem aspectos linguísticos, popularidade e características próprias do YouTube para reconhecer e prever o tipo do vídeo. A proposta foi validada em outros conjuntos de vídeos também extraídos do YouTube e os resultados revelaram uma precisão superior a 80%, sendo assim uma abordagem eficaz para o problema levantado.

Todos os trabalhos aqui listados utilizam algum tipo de análise textual de conteúdos para construir suas soluções, o que se assemelha de certa forma com o presente trabalho. Os três últimos trabalhos fazem uso dos dados do YouTube para suas análises, no entanto, suas propostas focam nos metadados dos vídeos do YouTube, ao contrário deste trabalho que utiliza os textos oriundos dos vídeos. Neles também estão presentes etapas de classificação manual dos dados para a análise, enquanto neste trabalho não existe a necessidade, pois a abordagem aqui usada (clusterização) trabalha sem a rotulação prévia dos dados. Sureka *et al.* (2010) tem como um dos objetivos encontrar perfis de usuários que se adéquem no contexto escolhido, já no presente trabalho o objetivo geral é encontrar qualquer tipo de perfil de *youtuber*, baseado na análise dos conteúdos dos vídeos. Assim como em Chau e Xu (2007) que procura entender os relacionamentos entre *blogs*, este trabalho procura interpretar os *clusters* obtidos após o processo de clusterização. E ao contrário de em Chaudhary e Sureka (2013), neste trabalho não existe uma etapa de classificação, e sim, somente encontrar grupos de *youtubers*. A Tabela 1 apresenta

[2] Rede de computadores infectados com *softwares* prejudiciais, sem o conhecimento de seus usuários.

a síntese da comparação dos trabalhos relacionados descritos neste capítulo com o trabalho proposto.

Tabela 1 – Comparação entre os trabalhos relacionados e o trabalho proposto

Autores	Clusterização	Fonte dos dados utilizada	Técnica utilizada
Chau e Xu (2007)	Não	Análise de relacionamentos entre blogs	Extração de informações e análise posterior
Sureka *et al.* (2010)	Não	Metadados de vídeos do YouTube	Extração de informações e análise posterior
Chaudhary e Sureka (2013)	Não	Metadados de vídeos do YouTube	Classificação
Aggarwal *et al.* (2014)	Não	Metadados de vídeos do YouTube	Mineração dos metadados
Trabalho proposto	**Sim**	**Legendas dos vídeos geradas automaticamente**	**Clusterização e interpretação dos resultados**

Fonte: Elaborado pelo autor.

4 METODOLOGIA

O presente trabalho busca descobrir e identificar perfis de canais ou *youtubers* do *site* de compartilhamento de vídeos *online* YouTube por meio do uso de técnicas de clusterização. O processo é feito primeiramente, por meio de sistemas de coleta de dados e processamento de forma automatizada, para finalmente a visualização dos resultados.

Assim sendo, a pesquisa tem como natureza do tipo aplicada, pois como definido em Prodanov e Freitas (2013): "Procura produzir conhecimentos para aplicação prática dirigidos à solução de problemas específicos". Baseado no objetivo de estudo, é possível defini-la como pesquisa exploratória, pois "visa proporcionar maior familiaridade com o problema, tornando-o explícito ou construindo hipóteses sobre ele"(PRODANOV; FREITAS, 2013). Ao categoriza-lá pelo procedimento técnico, é definida como pesquisa experimental, pois é determinado um objeto de estudo, são selecionadas as variáveis e definidas formas de controle e de observação dos efeitos (PRODANOV; FREITAS, 2013). Por fim, com relação a abordagem, a pesquisa possui características de pesquisa quantitativa, pois requer recursos e técnicas para traduzir em números os conhecimentos gerados pelo pesquisador (PRODANOV; FREITAS, 2013). Mas também possui características de pesquisa qualitativa, pois ambiente e fonte direta para coleta de dados, interpretação e atribuição de significados (PRODANOV; FREITAS, 2013). Portanto o ideal é classificá-la nos dois tipos, já que a pesquisa agrupa características de ambas.

4.1 Processo de coleta de dados

Inicialmente, os dados que serão usados na análise do sistema, ou seja, os textos das legendas de vídeos selecionados do YouTube, são obtidos por meio de requisições específicas a sua API. Por conta deste trabalho ser em colaboração com GRIM, os criadores podem classificados como pertencentes dos grupos de *Youtubers Gamers* e Infantis, já que as pesquisas do grupo focam nesses tipos de *youtubers*. Depois, foi usado como critério a popularidade, isso pode ser avaliado por meio do número de inscritos e visualizações. Todos os canais na qual foram coletados os textos possuem mais de 1 milhão de inscritos e vídeos com mais de 10 mil visualizações em média. Depois de selecionados os canais, foram então escolhidos os vídeos do mesmo.

No início da pesquisa, o conjunto total contava com 100 textos na qual foi mais importante a validação do funcionamento do mesmo, entender e otimizar o processamento

anterior do conjunto de dados de amostra (descrito na seção 4.2) do que o objetivo final do trabalho. No entanto, foram coletados mais textos para os testes deste trabalho, totalizando a quantidade de 300.

4.2 Processamento inicial dos textos

Para cada texto obtido no processo de coleta explicado na Seção 4.1 são feitos procedimentos para otimizar o conteúdo para as passos posteriores. Isto posto, primeiramente são removidos os caracteres especiais dos textos e todos os caracteres que compõem as palavras convertidos para letras minúsculas. No contexto dessa análise, é assumido que a capitalização das palavras não contém informações semânticas relevantes. Outro fator, é que as palavras presentes nos modelos do GloVe seguem essa configuração, desse modo é possível obter representações de mais palavras presentes nos textos.

Posteriormente ocorre a descoberta das palavras mais importantes do conjunto dos textos e remoção das *stop-words*, estas que são as palavras comuns em uma língua e que devem ser consideradas irrelevantes para o processo de análise de um texto em linguagem natural. O essencial para a análise final são as palavras mais importantes, que também, caracterizam o criador do vídeo em específico. Outro fator é que ao diminuir o número de palavras que serão analisadas, consequentemente reduz a carga de trabalho do sistema, pois menos objetos serão transformados em representações vetoriais e o processamento computacional será melhor direcionado.

A técnica utilizada para encontrar as palavras importantes e *stop-words* é a TF-IDF. Como explicado mais a fundo na seção 2.4, a técnica como medida estatística tem o intuito de identificar a importância de uma palavra de um documento em relação a um conjunto maior de documentos ou um corpus linguístico. Desta forma, além das *stop-words*, outras palavras, que por terem alta frequência no corpus e não serem consideradas importantes após a execução do TF-IDF, também serão removidas. Ao final dessa etapa, portanto, sobram apenas um conjunto de palavras características do corpus total dos textos.

4.3 Geração das representações vetoriais dos textos

Após o processamento explicado na seção anterior, deve ser obtidas as representações vetoriais de cada texto da amostra, pois elas serão necessárias no processo de clusterização

posterior. Foram utilizados os modelos gerados pelo algoritmo GloVe para obter essas representações. No entanto, por conta do GloVe contar apenas com representações vetoriais de palavras, em contrapartida com representações de textos da forma como é necessário para este trabalho, também foram efetuados procedimentos para converter esses vetores em somente um que serviu para o texto completo, descritos mais especificamente na seção 5.3.

4.4 Clusterização com o *k-means*

Nessa etapa ocorre o processo de obtenção dos agrupamentos, ou seja, é a clusterização em si. Os textos foram agrupados nos *clusters* de acordo com suas similaridades baseadas no cálculo das distâncias entre os objetos.

Como já definido neste presente trabalho, o k-means é um problema onde se tenta minimizar a Soma dos Erros Quadráticos e esse cenário pode ser atingido ao encontrar a quantidade ideal de *clusters* para serem divididos os dados. Para tal propósito foi empregado o Método *Elbow* para validar as execuções da clusterização. O método baseia-se na observação de que aumentar o número de *clusters*, pode reduzir a dispersão dos *clusters* (HAN *et al.*, 2011; RASCHKA; MIRJALILI, 2017). Portanto, foi executado o processo de clusterização em uma quantidade de vezes suficiente para um momento em que Soma dos Erros Quadráticos (ou inércia dos clusters) tenha a diferença mínima entre execuções da clusterização. Consequentemente, uma heurística para encontrar o número correto de *clusters* é usar o ponto de virada na curva da distorção com relação ao número de *clusters* (HAN *et al.*, 2011). Na subseção 5.4.1 é demostrado o uso do método.

4.5 Visualização dos resultados da clusterização

Por último, visualizar os resultados das análises é necessário para validar o processo e identificar os possíveis problemas. Essa parte pode ser feita simplesmente analisando a saída do sistema imprimindo os resultados inicialmente.

No entanto, para esse trabalho também foram utilizadas reproduções gráficas. Primeiramente para a visualização do resultado da clusterização após a etapa descrita na seção 4.4, onde foram plotados o posicionamento dos centróides dos *clusters* encontrados e também dos objetos que pertencem ao mesmo *cluster* do centróide. Esse método serve para compreender inicialmente os resultados, mas não é o desejável para uma análise completa.

O outro método utilizado é a geração de nuvens de palavras com as que foram as mais representativas dos *clusters*. As nuvens exibem um conjunto de palavras na forma de uma nuvem, o tamanho das palavras é totalmente proporcional a frequência que aparece no texto, possibilitando que com apenas uma visualização rápida identificar as principais palavras ou tópicos. Dessa forma, é possível ter um entendimento mais amplo e significativo dos resultados, pois utiliza uma representação mais clara ao ser humano.

5 EXPERIMENTAÇÃO E RESULTADOS

Neste capítulo serão abordados especificadamente os experimentos realizados baseados nos processos metodológicos discutidos no Capítulo 4. Como também, os respectivos resultados atingidos.

Como ambiente de desenvolvimento e realização dos experimentos, foi utilizada a linguagem de programação Python, na versão 3.6.9 para a implementação do sistema, em conjunto com a plataforma *online* de Google Colab[1].

Além disso, foram utilizadas as bibliotecas:

- *scikit-learn*[2] na versão 0.21.3 que disponibiliza as implementações de alguns algoritmos utilizados no sistema;
- *Matplotlib*[3] na versão 3.1.2 para geração dos gráficos para visualização dos resultados;
- *NumPy*[4] na versão 1.17.4 para operações entre matrizes e vetores;
- *NLKT* [5] na versão 3.2.5 para processamento dos textos;
- *wordcloud*[6] na versão 1.5.0 para criação das nuvens de palavras.

5.1 Processo de coleta de dados

Como já antecipado na seção 4.1, esta parte foi constituída pela coleta dos dados, mais especificamente os textos dos vídeos, para serem utilizados nos experimentos. Os textos foram provenientes de canais em que seus criadores podem ser classificados como pertencentes dos grupos de *Youtubers Gamers* e Infantis, isso de forma primitiva, sem o uso do sistema. A escolha destes grupos foi tanto por terem conteúdo característico, quanto pelo fato dos pesquisadores que farão a validação do sistema, além de possuir interesse nesse tipo de conteúdo, também já executaram processos manuais de classificação e agrupamento de perfis utilizando esses ou conjuntos de canais semelhantes. Para os experimentos preliminares foram utilizados 300 textos.

Nesse caso, os textos das legendas foram obtidas por meio da *YouTube Data API*[7],

[1] Disponível em <https://colab.research.google.com/>. Acesso em: 27 de Dez. de 2019.
[2] Disponível em <https://scikit-learn.org/stable/index.html>. Acesso em: 27 de Dez. de 2019.
[3] Disponível em <https://matplotlib.org>. Acesso em: 27 de Dez. de 2019.
[4] Disponível em <https://numpy.org>. Acesso em: 27 de Dez. de 2019.
[5] Disponível em <https://www.nltk.org>. Acesso em: 27 de Dez. de 2019.
[6] Disponível em <https://amueller.github.io/word_cloud/>. Acesso em: 27 de Dez. de 2019.
[7] Disponível em <https://developers.google.com/youtube/v3>. Acesso em: 27 de Dez. de 2019.

seguindo os seguintes passos:

1. Selecionar os identificadores únicos que representam cada vídeo individualmente. Eles ficam dispostos publicamente na *Uniform Resource Locator* (URL) específica de cada vídeo em seus parâmetros;
2. Configurar todo o projeto de autentificação que utiliza o protocolo *OAuth 2.0* para autorização e usar os serviços da API;
3. Executar a requisição *Hypertext Transfer Protocol* (HTTP) do tipo *GET* para o *endpoint* próprio da API: */captions*, com os três parâmetros obrigatórios. Primeiro, o identificador do vídeo, depois o tipo de resposta que deseja obter, que podem ser do tipo resumida e completa, e por último, a chave de acesso. A resposta é retornada no formato *JavaScript Object Notation* (JSON) e conta com uma lista com identificadores que representam as legendas de acordo com as localizações disponíveis para aquele vídeo;
4. Por último, executar outra requisição HTTP do tipo *GET* para o *endpoint /captions/[ID]*, onde o valor de *ID* deve ser o identificador da legenda que será feito o *download* e também como parâmetro, a chave de acesso.

Esses passos foram executados para cada um dos vídeos que foram usados nos experimentos do trabalho. Os textos também foram salvos para facilitar o acesso neste trabalho, quanto para outros que desejem utilizar desses dados.

5.2 Processamento inicial dos textos

A etapa de processamento dos textos é uma das mais importantes em todo o processo descrito neste trabalho, pois um bom trabalho de limpeza e preparação dos dados pode garantir e obter melhores resultados nas execuções das etapas posteriores. Para conseguir entender melhor os dados e bem como, descobrir os procedimentos necessários para tratamento inicial, foi usado um conjunto menor de textos inicialmente, para depois o conjunto final.

5.2.1 Letras minúsculas e "Tokenização"

O primeiro quesito utilizado para definir os procedimentos que foram realizados durante o processamento inicial dos textos foi a quantidade de palavras que eram retornadas representações vetoriais pelo GloVe. Por conta delas serem os itens principais para as formações

dos *clusters* pelo *k-means*, quanto mais representações encontradas, mais dados haverá para as análises. Por isso os caracteres das palavras foram transformados para minúsculos.

O outro procedimento foi separar as palavras de possíveis caracteres especiais que estivessem logo após (como no caso da expressão "Olá!"). Inicialmente, foram utilizadas expressões regulares para remover os caracteres especiais, no entanto, não foi encontrada uma expressão que se adequasse corretamente e não prejudicasse o conteúdo do texto. Assim sendo, foi alterado para ser executada a "tokenização" do texto usando a função própria da biblioteca *NLKT*. Isso foi feito para conseguir, no mínimo, a representação da palavra, pois apesar do GloVe contar com representações com os caracteres especiais, em alguns cenários não foi possível obtê-las e informações estavam sendo perdidas.

Na Tabela 2 é demonstrado os resultados após cada método desempenhado.

Tabela 2 – Quantidade de palavras que foram obtidas as representações vetoriais após procedimento realizado

Descrição do procedimento	Quantidade de vetores
Textos brutos	246.272
Após transformar caracteres para minúsculos	246.842
Após usar a "tokenização"	248.587

Fonte: Elaborado pelo autor.

5.2.2 Execução do TF-IDF

Após melhorar a obtenção das palavras diretamente do GloVe, os próximos passos feitos foram com o objetivo de melhorar a eficiência para a clusterização. Para tanto, foram diminuídos os textos individualmente para conter apenas as palavras consideradas importantes, removendo ruídos como as *stop-words* e caracteres de pontuação que não são considerados importantes para as análises dos conteúdos dos textos no contexto deste trabalho.

No sistema desenvolvido, foi utilizada a implementação para o TF-IDF disponibilizada pela biblioteca *scikit-learn*. Inicialmente, é feito o uso da classe *CountVectorizer* com o objetivo de converter o conjunto de textos para uma matriz com a contagem de cada termo individualmente. Cada coluna da matriz resultante representa uma palavra, cada linha representa um documento no conjunto de dados e os valores são a contagem em si. A classe foi configurada de forma a analisar individualmente cada palavra do corpus, selecionar 10 mil palavras para formar o vocabulário e ignorar os termos que possuem a frequência nos documentos do corpus acima de 30%. Além disso, também são adicionadas diretamente um conjunto de *stop-words* da

língua portuguesa previamente selecionadas e disponibilizadas pela biblioteca *NLKT*.

Por conseguinte a contagem dos termos, que também pode ser comparada a frequência dos termos no corpus, é calculado o IDF. Foi utilizada a classe *TfidfTransformer*, que assim como anteriormente, é implementada na biblioteca *scikit-learn*. A classe inicialmente computa os valores baseados na matriz de pesos de todo o corpus de documento, depois obtém os valores do TF-IDF de cada texto individualmente. Desta forma, os valores da matriz resultante serão as métricas do TF-IDF. Por fim, são selecionadas as 5 mil palavras mais importantes de cada texto separadamente, baseado nos valores do TF-IDF. Essas palavras são as que foram utilizadas nas análises realizadas nas etapas adiante.

5.3 Geração das representações vetoriais dos textos

Com a execução do processamento inicial dos textos, foram obtidas as representações vetoriais dos textos. Assim como a etapa anterior, esse procedimento é de grande importância para a análise, pois a partir dos resultados desta etapa que o processo de clusterização da etapa posterior é mais bem sucedido.

Portanto, para alcançar tal objetivo, primeiramente cada texto foi processado para obter as representações dos mesmos. Antes, são obtidas as representações vetoriais de cada palavra individualmente. Nessa tarefa são utilizadas as representações geradas pelo algoritmo GloVe, como explicado na subseção 2.3.1. O modelo produzido pelo algoritmo e usado nestes experimentos conta com vetores de 50 dimensões. Além disso, estas palavras também devem fazer parte do conjunto de palavras consideradas importantes de respectivo texto obtidas como descrito na seção 4.2.

Com as representações das palavras, é calculada a representação resultante do texto. O cálculo necessário é a média simples das representações das palavras do texto, ou seja, o somatório de todas as representações vetoriais obtidas do texto dividido pela quantidade de palavras. Por conta que todos os vetores possuem as mesmas dimensões, não se faz necessário redimensiona-los durante o processo de adição de dois vetores. A eficácia da abordagem de calcular a média simples é demostrada em trabalhos como o de Kenter *et al.* (2016) e Kenter e Rijke (2015).

Por fim, deve ser ressaltado que algumas palavras dos textos foram perdidas nesse processo, como já era de ser esperado pois mesmo que o GloVe possua uma quantidade de termos satisfatória (929.606 termos no modelo utilizado), ainda possui lacunas. Por isso, mesmo

que sempre sejam selecionadas as 5 mil palavras mais importantes daquele texto, não é garantido de existir uma representação vetorial para o termo.

Outro fator foi por conta dos textos serem gerados de forma automática pelo sistema do *YouTube*, algumas palavras não são transcritas em suas formas corretas. Isso foi o caso do termo "píncipe", por exemplo, que não existe na língua portuguesa, no entanto baseando-se no conteúdo e contexto do vídeo individualmente, é incontestável que se trata da palavra "príncipe". Consequentemente não foi encontrado no modelo do GloVe e possivelmente removido das palavras importantes na etapa do cálculo TF-IDF, mesmo se tratando de um termo que deve se fazer presente na análise. Para esse último caso descrito, não foi praticada uma solução, simplesmente as palavras que não eram transcritas corretamente, foram ignoradas.

5.4 Clusterização com o K-means

Para realizar a clusterização foi utilizado o algoritmo *k-means*, cujo o funcionamento está detalhado na subseção 2.2.1, que assim como no caso do TF-IDF, foi utilizada a implementação oferecida na biblioteca *scikit-learn*. A classe *KMeans* recebe em seu construtor o número de *clusters* que deve agrupar o conjunto de dados. A clusterização propriamente ocorre quando é executado o método que recebe como argumento o conjunto de dados, que são as representações vetoriais dos textos obtidas através do passo descrito na seção 5.3.

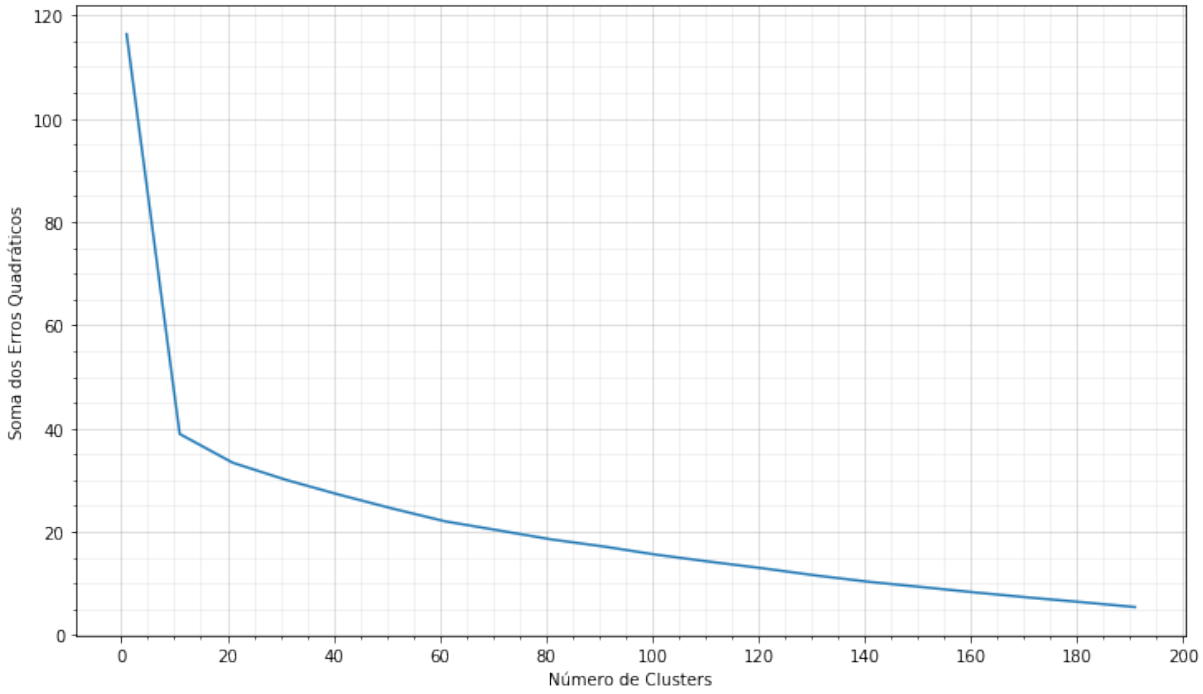
As intervenções feitas nesse passo são poucas, simplesmente o que já foi descrito. Isso se deve ao fato da eficiência do processo de clusterização depender mais dos procedimentos anteriores, que trabalham o conjunto de dados para transformar no melhor modelo possível para o algoritmo desta etapa. Portanto, durante os experimentos os valores de alguns parâmetros foram alterados até atingir os melhores resultados para a conclusão desta pesquisa. A título de exemplo, a quantidade de palavras do vocabulário gerado, ou o limiar da frequência dos termos explicado na subseção 5.2.2.

5.4.1 Validação dos resultados pelo Método Elbow

A fim de validar os procedimentos e encontrar o número ideal de *clusters*, foi executado o Método *Elbow* depois de cada mudança nos passos anteriores. Para melhor visualização foram plotados os resultados em função da Soma dos Erros Quadráticos dos *clusters*. No contexto do algoritmo utilizado, se refere a inércia dos *clusters*, disponível ao usar a classe *KMeans*.

Na Figura 2 são ilustrados os resultados ao definir o intervalo de 0 até 200 *clusters* para a clusterização com o conjunto de textos analisados. O gráfico contém a resultado do cálculo da Soma dos Erros Quadráticos no eixo vertical e a quantidade de *clusters* no eixo horizontal. De acordo com esse resultado, apesar de não exato, foi encontrado um ponto de grande variação na faixa dos 10 *clusters*. Com isso, na Figura 3 é demonstrado o resultado da clusterização utilizando $k = 10$. No entanto, é preciso ressaltar que o valor do erro não ficou estabilizado como deveria e os centróides dos agrupamentos ficaram próximos, o que pode significar que os *clusters* estão muito parecidos.

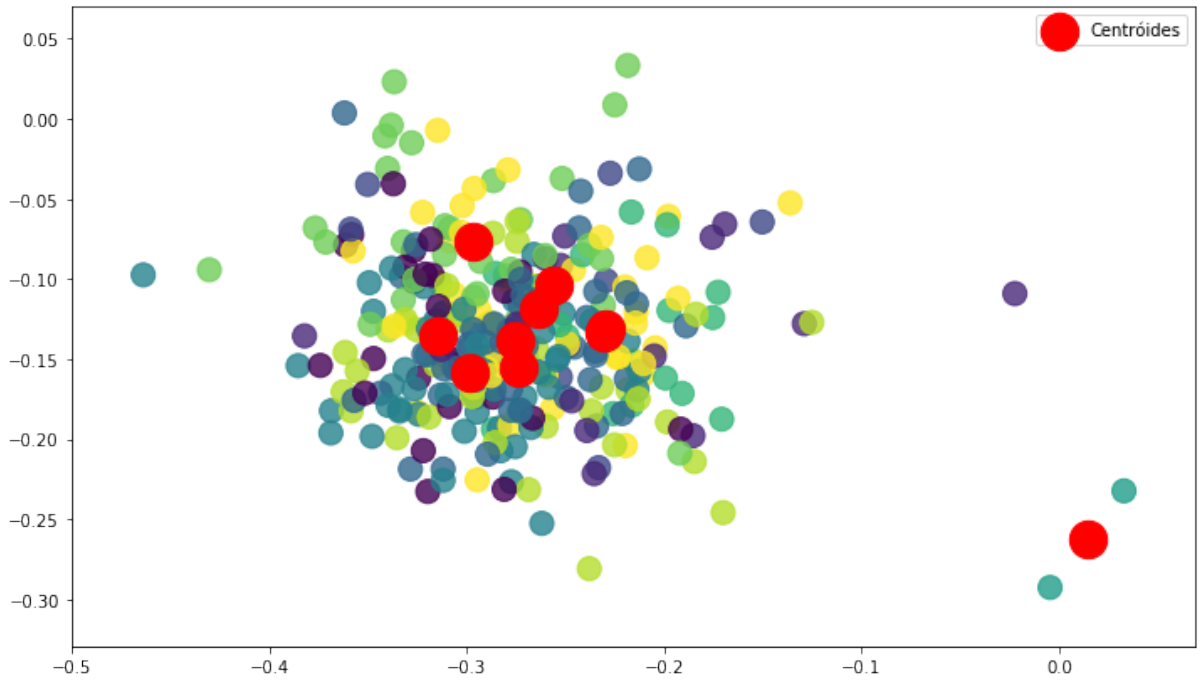
Figura 2 – Gráfico que demonstra o uso do Método *Elbow* no intervalo de 0 a 200 *clusters* na amostra de 300 textos



Fonte: Elaborado pelo autor.

Além disso, os agrupamentos gerados nem sempre eram os mesmos após cada clusterização mesmo mantendo a quantidade de *clusters*, os números ficaram semelhantes, variando em poucos objetos. No entanto, alguns agrupamentos foram repetidos, principalmente os que tinham menos elementos. Na Tabela 3 são expostas quantidades de objetos atrelados a cada *cluster* encontrado após a clusterização em que os resultados foram analisados e descritos nessa seção.

Figura 3 – Gráfico que demonstra a clusterização com o $k = 10$



Fonte: Elaborado pelo autor.

Tabela 3 – Quantidade de objetos em cada *cluster* após a clusterização

Identificador do *cluster*	Quantidade de objetos
0	27
1	51
2	8
3	33
4	2
5	25
6	18
7	45
8	30
9	46

Fonte: Elaborado pelo autor.

5.5 Visualização e análise dos resultados da clusterização

A fim de visualizar e entender com mais cuidado o resultado descrito na subseção 5.4.1, também foram feitas análises diretamente nos objetos dos *clusters* encontrados, com o intuito de entender melhor os resultados e validar manualmente. Para isso foram plotados gráficos de forma semelhante ao da Figura 3 para outras quantidades k de *clusters*.

A visualização básica dos resultados é por meio de gráficos gerados pela biblioteca *Matplotlib* e para a geração das nuvens de palavras diretamente via código foi utilizada a biblioteca *wordcloud*.

Outro procedimento foi a geração de nuvens de palavras dos textos que ficaram nos

mesmos *clusters* para comparar suas palavras mais frequentes manualmente. Apesar de alguns objetos no mesmo *cluster* não possuírem semelhanças aparentes, alguns casos foram encontradas semelhanças no conteúdo.

Nas Figuras de 4 a 6, são ilustradas nuvens de palavras que representam textos e suas respectivas palavras importantes selecionadas. Estão destacadas algumas palavras consideradas semelhantes e que podem ser usadas nos mesmos contextos. O *cluster* em que essas figuras foram geradas é o de número 5 contava com 25 objetos incluindo os usados para gerar as nuvens de palavras.

O vídeo da Figura 4 tem como título "BONECA LOL SURPRESA NA PENTEADEIRA COM AMIGA VAI PRA ESCOLA DE MOCHILA NOVA" e é do canal "Brinquedos e Bonecas". O vídeo existe uma narradora que conta uma história interpretada por bonecas. Durante o mesmo, também são descritos as roupas que a boneca está usando, bem como alguns adereços que podem ser colocados juntos. O vídeo também tem muito um discurso demonstrativo das bonecas como produto, também mostrando elas sendo retiradas das embalagens e quais assessórios vêm na caixa.

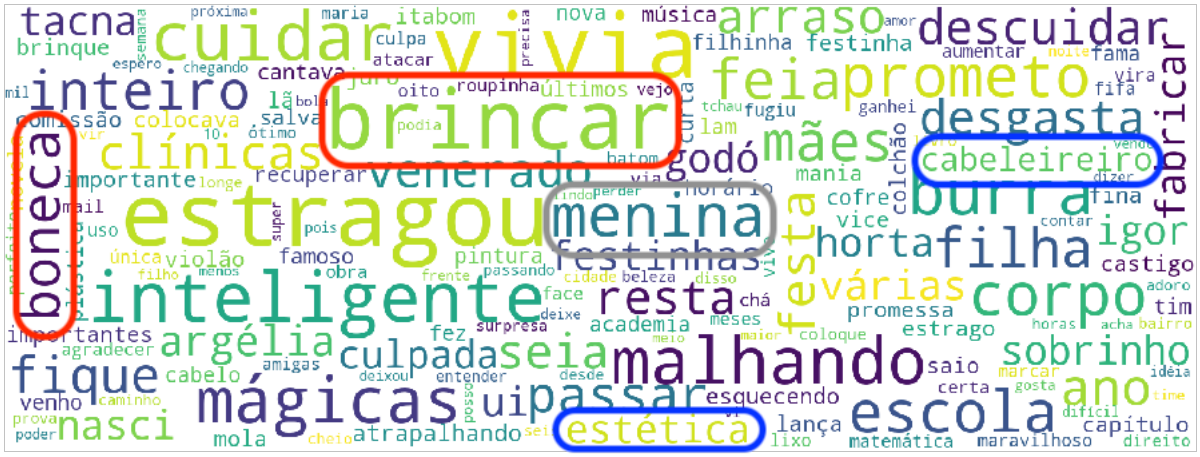
O segundo vídeo, no qual foi usado para gerar a nuvem de palavras da Figura 5 tem o título "PEITOS! | Sims 4 (2) - PupiGames" do canal "PupiGames". Ao contrário do primeiro canal, este é possui conteúdo *Gamer*, ou seja, relacionado a jogos digitais. Durante todo o vídeo a *youtuber* está jogando *The Sims 4* junto com uma colega e também narrando suas ações. No entanto, de forma mais específica, o vídeo se resume apenas a construção do personagem enquanto a dupla conversa sobre as escolhas de adereços estéticos para serem usados.

Apesar de contextos totalmente diferentes, os conteúdos se assemelham por tratarem principalmente de discussões sobre estética, no primeiro, das bonecas, e no segundo, da personagem do jogo sendo construída. Esse resultado demostra certa eficácia do sistema ao agrupar ambos objetos no mesmo *cluster*.

No entanto, no terceiro vídeo que originou a nuvem de palavras da Figura 6 a situação foi diferente já que apesar de contar com palavras semelhantes, o conteúdo não possui aspectos próximos. O vídeo tem o título "SITUAÇÕES CONSTRANGEDORAS NA ESCOLA" e é do canal "Julia Silva". A protagonista do vídeo é uma *youtuber* mirim e seus vídeos basicamente contam com conteúdos relacionados a vida dela, portanto mais direcionados ao público infantil. No vídeo em específico são contadas situações de vivências comuns de pessoas, o que não se pode encontrar paralelos com os outros vídeos. No entanto, assim como o segundo vídeo,

possui muitas palavras no diminutivo consideradas importantes pelo sistema, justificando assim a presença desse texto junto do segundo vídeo.

Figura 4 – Figura da nuvem de palavras do vídeo “BONECA LOL SURPRESA NA PENTEADEIRA COM AMIGA VAI PRA ESCOLA DE MOCHILA NOVA”



Fonte: Elaborado pelo autor.

Figura 5 – Figura da nuvem de palavras do vídeo “PEITOS! | Sims 4 (2) - PupiGames”

Fonte: Elaborado pelo autor.

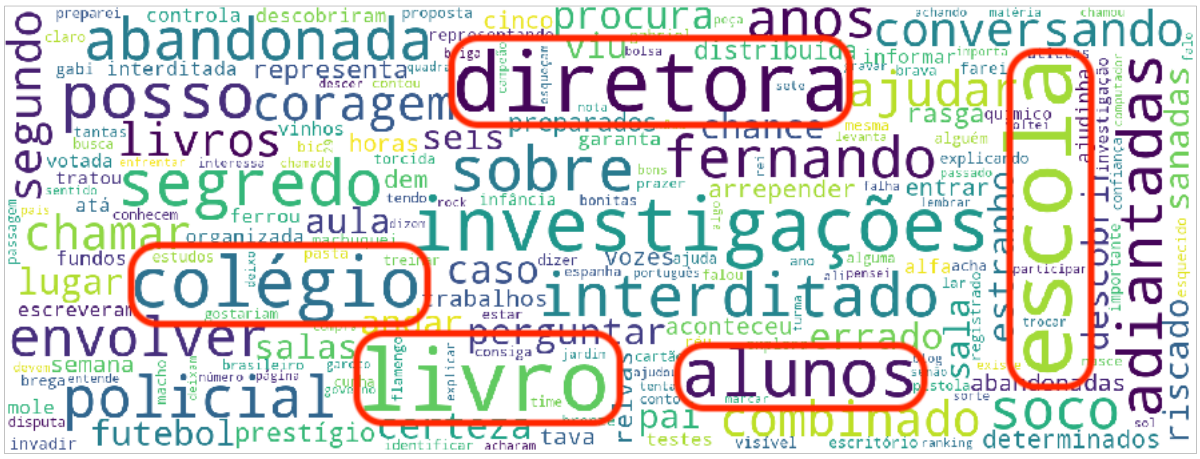
Ao analisar outro *cluster*, identificado com o número 9 e com 46 elementos, foi constatado que as palavras importantes de alguns vídeos indicam conteúdos relacionados ao ambiente escolar. Os vídeos estão ilustrados nas Figuras de 7 a 9 são “Minecraft Origens 8: CONVIDEI MEU AMIGO PARA PARTICIPAR DO SURVIVAL!” do canal “Jazzghost”, “O SIMSIMI É DO MAL!!!!!” e “MEU NOVO CRUSH! | Sims 4 (7) - PupiGames” ambos do canal “Pupigames”.

Entretanto, no caso desses três exemplos do *cluster*, nenhum dos vídeos é diretamente relacionado a escola ou algo parecido. Em todos, os *youtubers* estão jogando jogos diferentes

Figura 6 – Figura da nuvem de palavras do vídeo “SITUAÇÕES CONSTRANGEDORAS NA ESCOLA”

Fonte: Elaborado pelo autor.

Figura 7 – Figura da nuvem de palavras do vídeo “Minecraft Origens 8: CONVIDEI MEU AMIGO PARA PARTICIPAR DO SURVIVAL!”

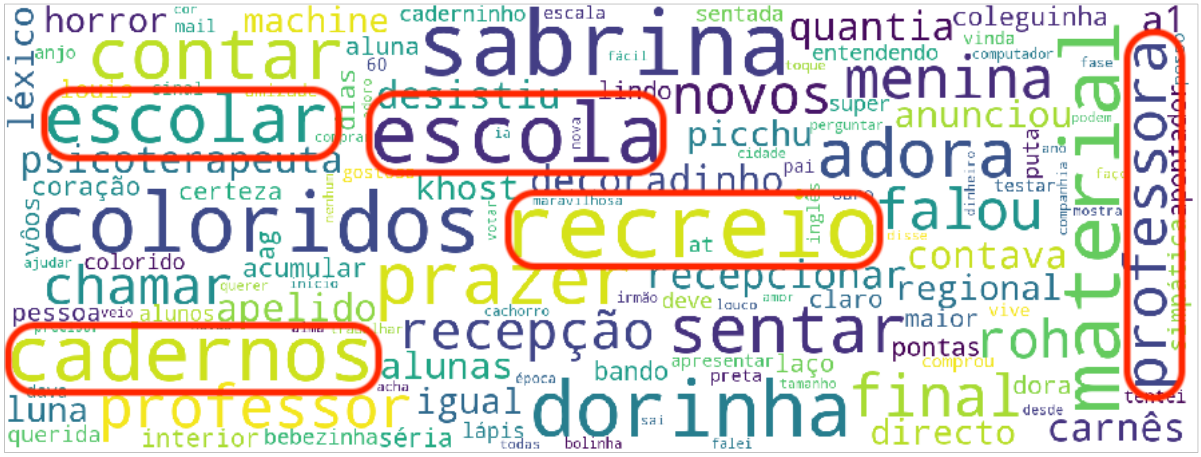


Fonte: Elaborado pelo autor.

Figura 8 – Figura da nuvem de palavras do vídeo “O SIMSIMI É DO MAL!!!!!”

Fonte: Elaborado pelo autor.

Figura 9 – Figura da nuvem de palavras do vídeo “MEU NOVO CRUSH! | Sims 4 (7) - PupiGames”



Fonte: Elaborado pelo autor.

e com cenários diferentes. Isso mostra que apesar de ser correto os objetos estarem juntos no mesmo grupo, por conta da similaridade de suas palavras, não quer dizer que o conteúdo também seja minimamente similar. Portanto, é notado que ao analisar as palavras isoladamente não é assegurado que será entendido o conteúdo do texto como um todo. O problema também se dá pelo fato da escolha das palavras mais importantes desses textos, o que é totalmente relacionado ao corpus total, o processo pode ficar prejudicado por não haver uma base tão grande. Também foi demonstrado que criadores podem estar presentes em mais de um *cluster*, pois seus vídeos podem contar com assuntos e temáticas diferentes comparado aos do próprio criador.

Em outro *cluster* encontrado, foram agrupados 18 elementos e três textos são ilustrados pelas Figuras de 10 a 12. Os dois primeiros vídeos são “MINHA LUA DE MEL! | Sims 4 (10) - PupiGames” do canal “PupiGames” e “MEU VIZINHO FICOU COMPLETAMENTE MALUCO! - ROBLOX (Hello Neighbor)” do canal “Filha Também Joga”. O terceiro é intitulado “PERGUNTAS IDIOTAS 11 ! Dicas de Fãs” do canal “Mileninha Stepanienco”, que não tem relação com os outros a princípio, já que são relacionados a jogos digitais. As palavras presentes nas nuvens de palavras, no entanto, expõem a similaridade de alguns termos relacionados a cozinha e comida, assunto este, é tratado de forma rápida e superficial em dois vídeos, e somente em um de forma mais presente. Novamente, apesar das palavras mais importantes serem similares, não quer dizer que elas representam com segurança a totalidade do conteúdo do vídeo em si.

Por último, ao analisar os dois *clusters* com menos objetos, primeiro com dois e o segundo com oito, não foram encontradas semelhanças importantes em seus objetos. Inclusive, o com somente dois elementos fica separado dos outros de acordo com o gráfico da Figura 3. Por isso, foi considerado estes *clusters* como *outliers*, ou seja, são valores aberrantes e atípicos.

Figura 10 – Figura da nuvem de palavras do vídeo “MINHA LUA DE MEL! | Sims 4 (10) - PupiGames”

Fonte: Elaborado pelo autor.

Figura 11 – Figura da nuvem de palavras do vídeo “MEU VIZINHO FICOU COMPLETAMENTE MALUCO! - ROBLOX (Hello Neighbor)”

Fonte: Elaborado pelo autor.

Figura 12 – Figura da nuvem de palavras do vídeo “PERGUNTAS IDIOTAS 11 ! Dicas de Fãs”

Fonte: Elaborado pelo autor.

6 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresenta um modelo de sistema que utiliza técnicas e algoritmos simples de clusterização para associar perfis de criadores de conteúdo para a plataforma do YouTube, e mesmo assim, atingir resultados aceitáveis. O procedimento aqui demostrado já pode ser utilizado de forma supervisionada por profissionais da área da comunicação para análises mais elaboradas do conteúdo de vídeos específicos. Além disso, os resultados trazem contribuições significativas que podem servir de base para pesquisas futuras na área e também para aprimoramento das técnicas para resolução deste problema abordado.

Sobre as dificuldades encontradas no decorrer do trabalho, são de diferentes níveis de complexidade sendo algumas inconvenientes para os resultados finais. Apesar dos sistemas de geração automática de legendas do YouTube serem bastante completos e adequados para a geração dos textos, ainda é possível perceber falhas no reconhecimento, principalmente relacionados a língua portuguesa. Por depender diretamente destes dados, falhas que prejudicam o conteúdo consequente atrapalham as análises desempenhadas nesta pesquisa.

Também relacionado ao YouTube, a base de dados precisou ser feita para ser utilizada no sistema. Por conta de limitações de acesso a API, o processo de coleta dos textos das legendas não pode ser otimizado da forma como era desejado, portanto limitado e dificultando a obtenção dos dados. Outro fator é que a base era inexistente antes do início deste trabalho, portanto um volume menor de dados foram utilizados.

Outra dificuldade que deve ser levantada é a limitação dos modelos de representações vetoriais disponíveis para língua portuguesa. No caso do modelo escolhido desenvolvido pelo NILC, as principais referências e fontes dos dados para o treinamento foram artigos acadêmicos e notícias, e não é possível afirmar que condiz com a forma cotidiana do brasileiro de se expressar, consequentemente não se adequando perfeitamente a dos *youtubers*.

A pesquisa demostra como principal resultado que *youtubers* diferentes possuem similaridades entre seus vídeos que podem ser captadas e reconhecidas apenas pelos textos e seus vídeos. Isso amplia as possibilidades de agrupamento e recomendação de criadores, pois considera o conteúdo dos vídeos no processo, algo ainda não acrescentado ao sistema de recomendação do YouTube.

A motivação deste trabalho surge no contexto de auxiliar nos projetos de pesquisas desempenhadas pelo GRIM que utilizam vídeos do YouTube para análises da língua falada e dos conteúdos abordados por criadores infantojuvenis. Na contemporaneidade, muitas tarefas podem

ser automatizadas e endereçados a sistemas computacionais inteligentes. Como demostrado nos resultados, a aplicação já possui um nível satisfatório de acertos e de total capacidade de servir como base para ser modificada para atingir resultados mais precisos.

Baseado nos resultados algumas modificações são necessárias para melhorar a assertividade do sistema apresentado. Primeiramente e mais principal modificação necessária a ser feita no sistema desenvolvido neste trabalho seria utilizar sentenças completas para as análises, pois é notório nos resultados que apenas as palavras individuais não expõem o conteúdo completo do texto.

Como trabalhos futuros, também pode-se destacar a contribuição dos resultados desta pesquisa e seus processos metodológicos como base para sistemas mais eficientes e otimizados para atacar a questão deste problema. Bem como, este pode ser considerado um primeiro passo para sistemas mais complexos que utilizem esse sistema de treinamento como base para um classificador de *youtubers* automatizado, que por exemplo, receberia um vídeo novo e de forma imediata trataria de coloca-lo dentro de um dos grupos existentes.

REFERÊNCIAS

AGGARWAL, N.; AGRAWAL, S.; SUREKA, A. Mining youtube metadata for detecting privacy invading harassment and misdemeanor videos. In: **2014 Twelfth Annual International Conference on Privacy, Security and Trust**. [S.l.: s.n.], 2014. p. 84–93.

ALEXA. **The top 500 sites on the web**. 2019. Lista dos 500 sites mais acessados no mundo. Disponível em: <https://www.alexa.com/topsites>. Acesso em: 28 mar. 2019.

ALLAIRE, J. J. **Using word embeddings**. 2018. il. color. Disponível em: <https://jjallaire.github.io/deep-learning-with-r-notebooks/notebooks/6.1-using-word-embeddings.nb.html>. Acesso em: 19 abr. 2019.

BENGIO, Y.; DUCHARME, R.; VINCENT, P.; JAUVIN, C. A neural probabilistic language model. **Journal of machine learning research**, v. 3, n. Feb, p. 1137–1155, 2003.

CHANG, S.-F.; MANMATHA, R.; CHUA, T.-S. Combining text and audio-visual features in video indexing. In: IEEE. **Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.** [S.l.], 2005. v. 5, p. v–1005.

CHAU, C. Youtube as a participatory culture. **New Directions for Youth Development**, v. 2010, n. 128, p. 65–74, 2010. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/yd.376>.

CHAU, M.; XU, J. Mining communities and their relationships in blogs: A study of online hate groups. **International Journal of Human-Computer Studies**, v. 65, n. 1, p. 57 – 70, 2007. ISSN 1071-5819. Information security in the knowledge economy. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1071581906001248>.

CHAUDHARY, V.; SUREKA, A. Contextual feature based one-class classifier approach for detecting video response spam on youtube. In: **2013 Eleventh Annual Conference on Privacy, Security and Trust**. [S.l.: s.n.], 2013. p. 195–204.

CHOLLET, F. **Deep Learning with Python**. 1st. ed. Greenwich, CT, USA: Manning Publications Co., 2017. ISBN 1617294438, 9781617294433.

COVINGTON, P.; ADAMS, J.; SARGIN, E. Deep neural networks for youtube recommendations. In: ACM. **Proceedings of the 10th ACM conference on recommender systems**. [S.l.], 2016. p. 191–198.

HAN, J.; KAMBER, M.; PEI, J. **Data Mining: Concepts and Techniques**. 3rd. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011. ISBN 0123814790, 9780123814791.

HARTMANN, N.; FONSECA, E.; SHULBY, C.; TREVISO, M.; RODRIGUES, J.; ALUÍSIO, S. Portuguese word embeddings: Evaluating on word analogies and natural language tasks. **CoRR**, abs/1708.06025, 2017. Disponível em: <http://arxiv.org/abs/1708.06025>.

JOHNSON, B.; SWENEY, M. **Google buys YouTube for $1.65bn**. Guardian News and Media, 2006. Disponível em: <https://www.theguardian.com/media/2006/oct/09/digitalmedia.googlethemedia>.

KENTER, T.; BORISOV, A.; RIJKE, M. D. Siamese cbow: Optimizing word embeddings for sentence representations. **arXiv preprint arXiv:1606.04640**, 2016.

KENTER, T.; RIJKE, M. D. Short text similarity with word embeddings. In: ACM. **Proceedings of the 24th ACM international on conference on information and knowledge management**. [S.l.], 2015. p. 1411–1420.

LEVY, O.; GOLDBERG, Y. Dependency-based word embeddings. In: **Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)**. [S.l.: s.n.], 2014. v. 2, p. 302–308.

MIKOLOV, T.; CHEN, K.; CORRADO, G.; DEAN, J. Efficient estimation of word representations in vector space. **arXiv preprint arXiv:1301.3781**, 2013.

MIKOLOV, T.; SUTSKEVER, I.; CHEN, K.; CORRADO, G. S.; DEAN, J. Distributed representations of words and phrases and their compositionality. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2013. p. 3111–3119.

MOBASHER, B.; DAI, H.; LUO, T.; NAKAGAWA, M. Discovery and evaluation of aggregate usage profiles for web personalization. **Data mining and knowledge discovery**, Springer, v. 6, n. 1, p. 61–82, 2002.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. In: **Empirical Methods in Natural Language Processing (EMNLP)**. [s.n.], 2014. p. 1532–1543. Disponível em: <http://www.aclweb.org/anthology/D14-1162>.

PRODANOV, C. C.; FREITAS, E. C. de. **Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico-2ª Edição**. [S.l.]: Editora Feevale, 2013.

RAMOS, J. *et al.* Using tf-idf to determine word relevance in document queries. In: PISCATAWAY, NJ. **Proceedings of the first instructional conference on machine learning**. [S.l.], 2003. v. 242, p. 133–142.

RASCHKA, S.; MIRJALILI, V. **Python Machine Learning, 2nd Ed.** 2. ed. Birmingham, UK: Packt Publishing, 2017. ISBN 978-1787125933.

REUTERS. **YouTube serves up 100 million videos a day**. CNET, 2006. Disponível em: <https://www.cnet.com/news/youtube-serves-up-100-million-videos-a-day/>. Acesso em: 27 dez. 2019.

SIMONET, V. Classifying youtube channels: a practical system. In: **Proceedings of the 2nd International Workshop on Web of Linked Entities (WOLE 2013), in Proceedings of the 22nd International conference on World Wide Web companion**. [s.n.], 2013. p. 1295–1304. Disponível em: <http://dl.acm.org/citation.cfm?id=2488164>.

SUREKA, A.; KUMARAGURU, P.; GOYAL, A.; CHHABRA, S. Mining youtube to discover extremist videos, users and hidden communities. In: . [S.l.: s.n.], 2010. p. 13–24.

YOUTUBE. **YouTube para a imprensa**. 2019. Dados estatísticos do YouTube. Disponível em: <https://www.youtube.com/intl/pt-BR/yt/about/press/>. Acesso em: 28 mar. 2019.

APÊNDICE A – LISTA DE CANAIS DO YOUTUBE

Neste apêndice são listados os canais de origem dos vídeos que foram extraídas as legendas analisadas neste trabalho.

Nome do Canal	**Quantidade de vídeos**
Manoela Antelo	20
Fran Nina e Bel para meninas	20
Julia Silva	20
Juliana Baltar	20
PupiGames	20
Planeta das Gêmeas	20
Jazzghost	20
Brinquedos e Bonecas	19
Baby Doll Kids	18
Mileninha Stepanienco	16
Tubalatudo	15
Show do Tiago	13
Planeta das Gêmeas Games	13
Paulinho e Toquinho	12
Kids Fun	11
LipaoGamer	10
Brincadeira de Criança	10
Brinquedos KidsToys Brasil	8
Filha Também Joga	8
FitDance Kids & Teen	5
Canal Bobinho Massinhas	1