JOÃO PEDRO SANTOS RODRIGUES

Um Método para a Construção de Grafos de Conhecimento a partir de Transcrições de Áudio

Dissertação de Mestrado apresentado ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná (PUCPR) como requisito parcial para obtenção do título de Mestre em Informática.

Curitiba
2021

JOÃO PEDRO SANTOS RODRIGUES

Um Método para a Construção de Grafos de Conhecimento a partir de Transcrições de Áudio

Dissertação de Mestrado apresentado ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná (PUCPR) como requisito parcial para obtenção do título de Mestre em Informática.

Área de Concentração: Ciência da Computação.

Orientador: Emerson Cabrera Paraiso.

Curitiba
2021

Sumário

Sumário i
Lista de Algoritmos iv
Lista de Figuras v
Lista de Tabelas vii
Abstract ix
Resumo x

Capítulo 1
Introdução 1
1.1 Motivação da Pesquisa . . . 3
1.2 Objetivos . . . 7
1.2.1 Objetivo Geral . . . 7
1.2.2 Objetivos Específicos . . . 7
1.3 Hipóteses de Pesquisa . . . 8
1.4 Contribuições Científicas . . . 8
1.5 Escopo do Trabalho . . . 9

Capítulo 2
Fundamentação Teórica 10
2.1 Processamento de Linguagem Natural . . . 10
2.1.1 Operações Básicas sobre Texto . . . 11
2.1.1.1 Tokenização . . . 11
2.1.1.2 Filtering . . . 11
2.1.1.3 Normalização de Palavras . . . 13
2.1.1.4 Encoding - Representação Vetorial . . . 14
2.1.2 Extração de Informações . . . 17
2.1.3 Topic Modeling . . . 19
2.1.3.1 Latent Dirichlet Allocation . . . 20
2.1.3.2 Avaliação de Tópicos Latentes . . . 22

2.2 Teoria dos Grafos . . . 23
2.2.1 Formalização de um Grafo . . . 23
2.2.2 Estruturas dos Grafos . . . 24
2.2.2.1 Formas de um grafo . . . 25
2.2.2.2 Atributos de um grafo . . . 26
2.3 Representação do Conhecimento . . . 27
2.3.1 Redes Semânticas . . . 28
2.3.2 Grafos de conhecimento . . . 28
2.3.2.1 Geração de Grafos de Conhecimento . . . 30
2.3.2.2 Refinamentos dos Grafos de Conhecimento . . . 31
2.3.2.3 Avaliação de Grafos de Conhecimento . . . 32

Capítulo 3

Trabalhos Relacionados 34

3.1 Sistemas de extração de informação a partir de redes sociais baseadas em vídeos . . . 35
3.2 Grafos de Conhecimento como Ferramentas de Representação do Conhecimento . . . 37
3.2.1 Sistemas de Q&A baseados em grafos de conhecimento . . . 38
3.3 Considerações Finais sobre o Estado da Arte e Lacunas de Pesquisa . . . 41

Capítulo 4

Procedimentos Metodológicos 43

4.1 Planejamento Inicial . . . 43
4.2 Coleta dos dados . . . 45
4.3 Desenvolvimento do Método . . . 47
4.4 Resultados e Avaliação . . . 49
4.4.1 Métricas de Análise de Grafos . . . 49
4.4.2 Métricas de Qualidade de Grafos . . . 49
4.4.3 Métrica de Análise de Tópicos Latentes . . . 50
4.4.4 Desenvolvimento de uma Base de Avaliação e Testes . . . 51

Capítulo 5

Método Proposto 52

5.1 Pré-processamento . . . 52
5.2 Processamento dos Metadados . . . 55
5.3 Extração da Valência e Subjetividade . . . 57

5.4 Extração dos Tópicos Latentes . . . 57
5.5 Extração das Entidades Nomeadas . . . 58
5.6 Segmentação de sentenças . . . 59
5.7 Extração das Informações Abertas (Tuplas de Conhecimento) . . . 61
5.8 Refinamento das Tuplas de Conhecimento . . . 62
5.8.1 Agrupamento de Tuplas Similares . . . 62
5.8.2 Seleção da Tupla mais Representativa de um Dado Grupo . . . 65
5.9 Geração do Grafo de Conhecimento . . . 66

Capítulo 6

Resultados 68
6.1 Obtenção de uma base de dados . . . 68
6.2 Um método de modelagem de tópicos em transcrições de vídeos do Youtube 71
6.3 Um método para extração e refinamento de tuplas de conhecimento . . . 74
6.4 YouGraph e suas aplicações práticas . . . 81
6.4.1 Um método para identificação de conteúdo patrocinado em vídeos do Youtube . . . 82
6.4.2 Uma análise das principais características que tornam um vídeo influente (viral) . . . 86

Capítulo 7

Conclusão 95

Capítulo 8

Anexos 97
8.1 Tabela Entidades Nomeadas . . . 97
8.2 Stop Words Complementares . . . 98
8.3 Palavras-Chave Patrocínio . . . 98
Referências Bibliográficas 99

Lista de Algoritmos

1 Método proposto para a seleção dinâmica do melhor modelo LDA 58
2 Agrupamento de tuplas similares a partir de aprendizado não supervisionado; 64
3 Método proposto para agrupar tuplas similares através da distância de cosseno 65

Lista de Figuras

2.1 Processo de Tokenização Fonte: Spacy - <https://spacy.io/usage/spacy-101> 12
2.2 Exemplo de corpus com 3 documentos 14
2.3 Representação do One Hot Encoding 14
2.4 Representação com contagem de frequência 15
2.5 Representação com contagem de frequência 15
2.6 Arquitetura CBOW vs Skip-Gram. Fonte: (MIKOLOV et al., 2013b) . . . 16
2.7 Noticia veiculada no G1. Fonte: https://bityli.com/v1rZE 20
2.8 Framework do LDA. Fonte: (BLEI; NG; JORDAN, 2003) 21
2.9 À esquerda uma ilustração de Königsberg e à direita a sua representação em grafos. Adaptado de: (GRIBKOVSKAIA; SR; LAPORTE, 2007) 23
2.10 Exemplo de um digrafo (à esquerda) e de um grafo não direcionado (ou dirigido) à direita 24

3.1 String de busca utilizada 35
3.2 Principais componentes de um sistema de Q&A. Fonte: (UNGER; FREITAS; CIMIANO, 2014) 39

4.1 Estrutura da Pesquisa 43
4.2 Subgrafos do GC 47
4.3 Metodologia para obtenção do padrão ouro. 51

5.1 Método Proposto 52
5.2 Processo do pré-processamento 53
5.3 Demostração do Sub-Grafo dos Metadados 56
5.4 Extração dos tópicos latentes 59
5.5 Geração das pseudo sentenças. 61
5.6 Geração das pseudo sentenças. 61

5.7 Exemplo de um grafo de conhecimento no Neo4j 67

6.1 Total de vídeos obtidos via ASR e transcrição manual na base PT-BR . . . 70
6.2 Distribuição dos tópicos criados pelos canais 72
6.3 Tree Map e Wordcloud gerado pelo LDA 73
6.4 Total de sentenças distribuídas por extratos 77
6.5 Obtenção do padrão ouro a partir dos agrupamentos individuais. 78
6.6 Percentuais de falhas dos tag métodos *embedding_distortion* e *tsne_distortion*. 80
6.7 Gráfico de diferenças críticas para coeficientes de similaridade obtidos pelos diferentes Tag Métodos. 81
6.8 Método proposto para extração de conteúdo patrocinado. 83
6.9 Processamento das transcrições manuais. 84
6.10 Processamento das transcrições automáticas. 85
6.11 Exemplo da obtenção das *pseudo-sentenças*. 85
6.12 Resultados obtidos com a execução do método de detecção de vídeos patrocinados. 86
6.13 Equação da regressão binomial não negativa gerada 87

Lista de Tabelas

1.1 Exemplos de transcrições ASR e manual. 6
1.2 Alguns erros de speech recognition das transcrições 6

3.1 Trabalhos encontrados no mapeamento sistemático e não citados nesta seção. 42

5.1 Exemplos de NERs extraídas e sua representação na forma de tupla 60
5.2 Exemplo de tuplas de conhecimento extraídas a partir de um extrator de informações aberto. 62

6.1 Total de vídeos extraídos por categoria. Base Inglês 69
6.2 Total de canais e vídeos extraídos por categoria. Base PT-BR 70
6.3 Exemplos de tópicos pouco representativos no modelo de k=80. 72
6.4 Tópicos Rotulados 88
6.5 Principais tópicos criados *versus* categorias de origem dos canais 89
6.6 Tuplas extraídas de um vídeo. 89
6.7 Exemplos de tuplas extraídas 89
6.8 Exemplo de tuplas extraídas e sem semântica 90
6.9 Alguns erros de speech recognition das transcrições 90
6.10 Métodos de agrupamento de tuplas similares 90
6.11 Coeficiente de similaridade entre *cluster ensembles* e avaliadores humanos. 91
6.12 Coeficiente de similaridade entre *cluster ensembles* e avaliadores humanos considerando limiares de similaridade a partir de 75%. 91
6.13 Exemplo de sentenças e pseudo-sentenças obtidas 92
6.14 Exemplos de Tuplas Extraídas - Transcrição Manual 92
6.15 Exemplos de Tuplas Extraídas - Transcrição ASR 93
6.16 Exemplos de Tuplas Extraídas - Transcrição ASR 93
6.17 Variáveis contínuas e estatística descritiva 93

6.18 Variáveis categóricas e estatística descritiva 94

8.1 Traduzido de: https://spacy.io/api/annotation 97
8.2 Lista de Stop Words Expandida. 98
8.3 Lista de palavras-chave candidatas à patrocínio. 98

Abstract

Social networks are already part of people's daily lives. Social networks, such as Facebook, Twitter, Instagram, and Youtube, among others, allow the interaction, communication, and socialization of millions of people daily. Currently, one of the most successful social networks is Youtube with more than 2 billion hours watched daily. Given the dimensions of this platform, it is natural to understand that Youtube has become an essential part of companies' digital marketing strategies, with massive investments being made by these companies. Given this context, we propose the development of a method capable of acquiring knowledge from audio transcripts of Youtube videos, together with their metadata (title, description, number of views, ...). All learned knowledge will be represented in the form of a knowledge graph. Knowledge graphs are an interesting structure for the representation of knowledge, as they allow the representation of knowledge from concepts and entities. Based on the method developed, it was possible to detect sponsored content in videos, identify virality levels (digital engagement) and automatically classify the subjects present (topic modeling) in the transcripts. Still in this context, the method presented in this work could serve as an important management tool in marketing for the detection of trends (trend topics), detection of virality in videos, analysis of synergy between brands in digital influencers, in addition to other applications. Finally, this research also developed two public datastes with audio transcripts from Youtube, which together have more than 50,000 videos and metadata.

Keywords: Knowledge Graph; Video Transcription; Information Extraction

Resumo

As redes sociais já fazem parte do dia a dia das pessoas. Redes sociais, como o Facebook, Twitter, Instagram, Youtube, dentre outras, permitem a interação, comunicação e socialização de milhões de pessoas diariamente. Atualmente, uma das redes sociais de maior sucesso é o Youtube com mais de 2 bilhões de horas assistidas diariamente. Dada as dimensões desta plataforma, é natural entendermos que o Youtube tornou-se uma parte essencial nas estratégias de *marketing digital* das empresas, com massivos investimentos sendo feitos por estas companhias. Dado este contexto, propomos, o desenvolvimento de um método capaz de realizar a aquisição de conhecimento a partir das transcrições de áudio de vídeos do Youtube, juntamente com os seus metadados (título, descrição, número de views, ...). Todo o conhecimento aprendido será representado sob a forma de um grafo de conhecimento. Os grafos de conhecimento são uma interessante estrutura para a representação do conhecimento, por permitir a representação do conhecimento a partir de conceitos e entidades. A partir do método desenvolvido, foi possível realizar a detecção de conteúdos patrocinados em vídeos, a identificação de teores de viralidade (engajamento digital) e a classificação automática dos assuntos presentes (modelagem de tópicos) nas transcrições. Ainda neste contexto, o método apresentado neste trabalho poderia ainda servir como uma importante ferramenta gerencial em marketing para detecção de tendências (*trend topics*), detecção de viralidade em vídeos, análise de sinergia entre marcas em influenciadores digitais, além de outras aplicações. Por fim esta pesquisa também desenvolveu duas bases públicas com transcrições de áudio do Youtube onde somadas possuem mais de 50 mil vídeos e seus metadados.

Palavras-chave: Grafos de Conhecimento; Transcrição de áudio; Extração de informação

Capítulo 1

Introdução

A descoberta de conhecimento em bases de dados (*Knowledge Discovery in Databases* - KDD) é um importante campo da Ciência da Computação que tem como objetivo processar dados permitindo assim sua interpretação com o intuito de detectar tendências e padrões previamente desconhecidos (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996). O KDD permite o desenvolvimento de ferramentas que auxiliam na tomada de decisão. Os primeiros estudos em KDD iniciaram-se na década de 80 e focavam quase que exclusivamente no processamento de dados estruturados (bases de dados relacionais). Entretanto, atualmente existem inúmeros trabalhos e pesquisadores que focam seus esforços no processamento de dados não estruturados. Existem diversas fontes de dados não estruturadas disponíveis e, entre elas, pode-se citar: arquivos de áudio, textos de redes sociais e imagens em sites web, repositórios particulares, entre outros. Cada tipo de fonte de dado não estruturado possui suas próprias características e peculiaridades e o seu processamento tende a ser uma tarefa não trivial (REZENDE, 2003).

O Processamento de Linguagem Natural (PLN) é a área da Inteligência Artificial responsável por processar dados textuais. Trata-se de uma subárea de pesquisa multidisciplinar que contempla a Linguística, a Estatística, além da Inteligência Artificial. Segundo (JURAFSKY, 2000) o PLN permite a extração de conceitos e significados (semântica) através da análise sintática, morfológica e pragmática de arquivos textuais. Desta forma, pode-se transformar dados não estruturados em estruturados utilizando algoritmos de extração de informação (MARTINEZ-RODRIGUEZ; HOGAN; LOPEZ-AREVALO, 2018).

Para que seja possível a utilização das informações extraídas de dados não estruturados e, consequentemente, gerar conhecimento (atividade conhecida como aquisição de conhecimento), é importante armazenar estas informações em alguma estrutura de dados que permita realizar análises e inferências. Em outras palavras, significa gerar um modelo de representação de conhecimento ou modelo conceitual. Podemos entender um modelo

conceitual como uma abstração ou uma representação de parte da realidade que está presente na mente das pessoas (GUIZZARDI, 2005). Estes modelos também são chamados de bases de conhecimento por alguns autores (RUSSELL; NORVIG, 2016). Existem diversas abordagens para a representação do conhecimento, dentre as quais podemos citar as ontologias (FUNG; BODENREIDER, 2019), taxonomias (GANI et al., 2016), folksonomias (WAL, 2009), frames (MINSKY, 1974) e os grafos de conhecimento (GC) (PAULHEIM, 2017).

Apesar dos grafos de conhecimento serem uma forma de representação do conhecimento relativamente nova (o termo foi difundido pela Google em 2012 (SINGHAL, 2012)), seu uso atraiu grande interesse da indústria e da comunidade científica nos últimos anos (JI et al., 2020). Seu desenvolvimento está intimamente ligado à definição da web semântica (BERNERS-LEE; HENDLER; LASSILA, 2001), que prega que, páginas web, *tags* e as próprias palavras presentes nos documentos são conceitos com significados específicos. Desta forma ao se realizar uma consulta[1] por estes recursos (chamada literalmente de "coisa" - thing), a busca processa os conceitos envolvidos, e não somente o texto bruto (strings). Esta mudança de paradigma, ao se buscar conceitos e não palavras-chave isoladas, pode resolver uma série de dificuldades antigas da área da recuperação da informação e PLN, como por exemplo:

- ambiguidade: ao se analisar o conceito e não a *string*, é possível inferir o domínio da consulta de entrada e com isso determinar qual é o melhor conceito que responde a busca realizada;
- sumarização: relacionando os conceitos de uma consulta de entrada juntamente com a extração de informação dos textos da base de dados é possível gerar sumarização de textos de forma confiável (WU et al., 2018);
- co-relacionamento entre conceitos: por meio do formalismo do RDF[2] é possível expandir um determinado GC específico com outros GCs de diferentes fontes e domínios (este processo é conhecido como linked data);

Segundo Hogan et al. (2021), os grafos de conhecimento podem ser separados em duas categorias principais: os Open Knowledge Graphs e os Enterprise Knowledge Graphs. Os Open Knowledge Graphs são bases públicas e disponíveis para o uso em geral. Eles podem ser mantidos por uma comunidade de voluntários e/ou utilizando dados extraídos

[1] O termo *query* é comumente utilizado como sinônimo de busca ou consulta no âmbito de banco de dados e grafos de conhecimento.

[2] RDF é uma metalinguagem que tem como objetivo formalizar a representação de informações na Internet através de um identificador único (mais detalhes podem ser encontrados na seção 2.1.2).

da Wikipedia. Existem diversos GCs públicos disponíveis, e entre os mais proeminentes podemos citar: o DBpedia (LEHMANN et al., 2015), Freebase (BOLLACKER et al., 2007), Wikidata (VRANDEČIĆ; KRÖTZSCH, 2014) e YAGO (HOFFART et al., 2011). Já os Enterprise Knowledge Graphs, são bases de conhecimento proprietárias mantidas para fins internos de gestão da empresa (contendo dados e regras de negócios específicas) ou com propósitos comerciais. Um exemplo são os motores de busca da Google e Bing (SINGHAL, 2012; SHRIVASTAVA, 2017).

Desta forma, os grafos de conhecimento se configuram como uma interessante estrutura para a representação do conhecimento. Entretanto, os GCs possuem algumas particularidades que devem ser consideradas no seu uso. Primeiramente, por definição os grafos de conhecimento são sempre considerados incompletos e com possíveis erros lógicos (HOGAN et al., 2021), (PAULHEIM, 2017), (PUJARA et al., 2013). Como os GCs podem consumir diferentes fontes dados, além de dispensar de um esquema lógico formal e pré-definido, é natural e esperado, a presença de dados inconsistentes (HOGAN et al., 2021). Para tratar esta questão, os grafos de conhecimento devem ser capazes de: (i) encontrar estas possíveis contradições (atividade conhecida como avaliação de GCs) e (ii) corrigir estes erros (tarefa conhecida como refinamento de GCs).

1.1 Motivação da Pesquisa

As redes sociais já fazem parte do dia a dia das pessoas. Redes sociais, como o Facebook, o Twitter, o Instagram, o Youtube, dentre outras, permitem a interação, comunicação e socialização de milhões de pessoas diariamente (AGGRAWAL et al., 2018). Segundo os autores em (KHAN; VONG, 2014), a forma que criamos, consumimos e compartilhamos informações mudou. Nunca na história geramos tanta informação e em tão pouco tempo.

Uma das redes sociais mais populares é o Youtube (plataforma de criação e compartilhamento de vídeos mantida pela Google) com aproximadamente 2 bilhões de horas de vídeos assistidas diariamente e com mais de dois bilhões de usuários ativos na rede (YOUTUBE, 2020). Diversos fatores podem explicar o sucesso do Youtube, como, por exemplo, o seu formato voltado ao compartilhamento de vídeos de forma gratuita. Além disso, ele disponibiliza uma plataforma de monetização dos vídeos criados pelos produtores através de anúncios patrocinados, ou seja, apesar do Youtube ser uma plataforma gratuita para os usuários finais, ele ainda permite a geração de renda por parte dos produtores de conteúdo. Os conteúdos produzidos possuem assuntos diversificados desde *gameplays* de jogos online, dicas de moda e beleza, chegando a aulas e cursos online.

Desta forma, dada as dimensões desta plataforma, é natural entendermos que o Youtube se tornou uma parte essencial nas estratégias de *marketing digital* das empresas (ARORA et al., 2019). Devido a proximidade que os influenciadores digitais possuem com as suas audiências, eles possuem uma grande relevância no processo de formação da opinião do seu público alvo, de forma que as empresas dependem cada vez mais da promoção de influenciadores digitais para apresentar, promover e anunciar seus produtos e serviços (LI; SHI; WANG, 2019). Somente em 2020, foi previsto um investimento de 11,76 bilhões de dólares em campanhas de marketing direcionadas ao Youtube (CLEMENT, 2019). Entretanto, os massivos investimentos em campanhas publicitárias e principalmente a condução de políticas agressivas em tornar suas marcas e produtos evidentes nos meios digitais, acabam por trazer um efeito colateral indesejado: uma sobrecarga de conteúdos em cima dos consumidores (LEE; HOSANAGAR; NAIR, 2018). Indo além, a própria abundância de conteúdos disponíveis, mesmo sem vínculos comerciais, dificulta a diferenciação dos próprios influenciadores nos meios digitais, de forma que o processo de engajamento e retenção dos públicos alvo se torna comprometido. De acordo com o trabalho de (GAUSBY, 2015), as pessoas levam em torno de 8 segundos para determinar se um conteúdo é relevante ou não.

Assim sendo, existe um grande interesse econômico e acadêmico na compreensão dos seguintes itens: quais caraterísticas favorecem o engajamento do público; quais atributos tornam uma pessoa influenciadora; e principalmente quais conteúdos estão sendo difundidos por estes influenciadores. No âmbito do engajamento do público, diversos autores estudaram o envolvimento das audiências a partir de postagens do Facebook (BANERJEE; CHUA, 2019; HUGHES; SWAMINATHAN; BROOKS, 2019; SABATE et al., 2014) e do Twitter (FRANCALANCI; HUSSAIN, 2017). Em relação aos aspectos do que torna uma pessoa influente ou não, em (BALABANIS; CHATZOPOULOU, 2019) os autores exploram como determinadas características comportamentais e comunicacionais alteram a percepção das pessoas durante o consumo de um dado conteúdo. Já em relação a análise do conteúdo publicado a partir de redes sociais (processo também conhecido como recuperação da informação), houve extensivos estudos a partir do Facebook (NOY et al., 2019), Twitter (SENEVIRATNE; SAUER; ROTH-BERGHOFER, 2013) e LinkedIn (HE; CHEN; ARGAWAL, 2016).

Em relação às redes sociais baseadas em vídeos (como por exemplo o Youtube), diversas pesquisas foram propostas com o objetivo de adquirir e analisar novos conhecimentos (KAUSHIK; SANGWAN; HANSEN, 2013), (RANGASWAMY et al., 2016), (WÖLLMER et al., 2013), (GERHARDS, 2019) e (SCHWEMMER; ZIEWIECKI, 2018). Entretanto, apesar dos autores utilizarem uma rede social baseada em vídeos como fonte

de dados em seus processos, tais pesquisas objetivaram o desenvolvimento ou de ferramentas de análise de sentimento, ou de sistemas de recomendação, ou ainda na detecção de tendências a partir do título e descrição dos vídeos. Sendo assim, foi desprezada uma importante fonte de dados que são as transcrições de áudios apresentados em uma publicação. Este fato torna a exploração deste tipo de dado uma interessante lacuna de pesquisa a ser explorada.

Como justificativa às poucas pesquisas estudando as transcrições de áudios do Youtube (e outras redes sociais baseadas em vídeos) podemos citar:

- Complexidade deste tipo de dado;
- Priorização no estudo de outras redes sociais em detrimento do Youtube;

Trabalhar com transcrições de áudio em vídeos não é uma tarefa trivial. No caso do Youtube, as transcrições de áudio podem ser geradas de duas formas diferentes: (i) a partir da anotação manual do áudio[3]; e (ii) a partir de algoritmos que façam o reconhecimento automático da fala (*automatic speech recognition* (ASR))[4]. O texto gerado a partir do ASR pode possuir ruídos, como erros sintáticos e semânticos; com isso, o uso deste dado bruto em um sistema de aquisição de conhecimento pode ser prejudicado. Além desta dificuldade, devido ao fato das transcrições serem representações do que foi falado pelo autores nos vídeos, elas naturalmente possuem um tom mais coloquial que também dificulta o seu uso (WÖLLMER et al., 2013). Na Tabela 1.1, temos como exemplo, dois trechos transcrições de áudio obtidos de dois vídeos do Youtube. No primeiro exemplo, temos um exemplo de transcrição ASR e no segundo manual. Na transcrição ASR, podemos notar que os substantivos próprios estão escritos em letras minúscula e também a ausência de pontuações. Também podemos notar que estes textos possuem um tom mais informal e com a presença de gírias. Já na Tabela 1.2 temos alguns exemplos de transcrições ASR geradas de forma incorreta.

Dada estas dificuldades, o interesse dos pesquisadores neste tipo de dado foi limitado, de forma que foram priorizados estudos explorando dados do Facebook e Twitter, já que estes fornecem uma API pública para consulta e extração de dados.

Finalmente, este estudo propõe o desenvolvimento de um método capaz de realizar a aquisição de conhecimentos a partir das transcrições de áudio de vídeos do Youtube, juntamente com os seus metadados (título, descrição, número de *views*, *etc*...). Todo o conhecimento aprendido será representado sob a forma de um grafo de conhecimento.

[3]Youtube help. Disponível em <https://support.google.com/youtube/answer/2734796?hl=en>
[4]Youtube help. Disponível em <https://support.google.com/youtube/answer/6373554?hl=en>

Tipo da Transcrição:	**Transcrição:**	**Url:**
ASR	... se liga na galerinha que a partir dessa lista pra começar nossa lista a gente vai falar primeiro de um personagem na cabo da série de filmes ...	https://www.youtube.com/watch?v=6qPt12ebyaA
	... passa a atacar os jovens inclusive johnny depp que nem lembrava que está nesse filme só ...	
Manual	Cara, não tem nada mais mutante que o passado. O passado tá sempre se transformando. O passado é movediço, é cambiante, a gente nunca pode confiar no passado! ...	https://www.youtube.com/watch?v=-aTBmEDujYo
	... É tu, na tua ignorância, que não sabe! Mas tu vai saber. Vem vem, vem conhecer esse presidente! ...	

Tabela 1.1: Exemplos de transcrições ASR e manual.

A biologia do Baby Yoda | Nerdologia **Link:** https://www.youtube.com/watch?v=0JkqXpefUt4		
Tempo do Vídeo:	**Transcrição Obtida:**	**Texto Correto:**
5:12	... nosso cérebro para falar **com vivem** grandes sociedades ...	... nosso cérebro para falar, **conviver em** grandes sociedades ...
5:20	... agora se o **iodo** é tão inteligente que ele precisa de décadas para ...	... agora se o **Yoda** é tão inteligente que ele precisa de décadas para ...
6:01	... que demanda menos batimentos por segundos a gente podia inclusive **vê** ainda mais...	.. que demanda menos batimentos por segundos a gente podia inclusive, **viver** ainda mais...
7:13	...equivalente **a continuado** ficaria na frente **jorge** rodrigues foi...	...equivalente **ao acolchoado,** ficaria na frente. **Daniel** Rodrigues foi...

Tabela 1.2: Alguns erros de speech recognition das transcrições

Diversas aplicações poderiam se beneficiar destas implementações, dentre as quais pode-se citar sistemas de curadoria de informações, sistemas de perguntas e respostas (do inglês *question and answering system* - Q&A), motores de inferência, entre outros.

Neste contexto, o método apresentado neste trabalho poderia ainda servir como uma importante ferramenta gerencial em marketing para detecção de tendências (*trend*

topics), detecção de viralidade em vídeos, análise de sinergia entre marcas em influenciadores digitais, além de outras aplicações. Isto seria feito primeiramente extraindo tuplas de conhecimento a partir das transcrições de áudio, a partir de um extrator de informações. Com isto, cada conhecimento extraído (na forma de tuplas de conhecimento) seria vinculado em um grafo, permitindo o mapeamento dos conceitos extraídos entre diferentes vídeos e autores. Juntamente com demais meta dados presentes nos vídeos (como o número de visualizações, curtidas e comentários), o grafo gerado pode ser utilizado para montar "redes de influências", onde seria possível o mapear e inferir o engajamento de determinados conteúdos a partir de um influenciador digital, ou ainda mapear as opiniões de uma dada audiência a partir de seus comentários. Por fim, utilizando uma abordagem similar a utilizada em (CIAMPAGLIA et al., 2015), seria possível validar os fatos representados na base através da similaridade semântica dos nós, e ao analisar o fator de impacto dos recursos de origem (autor de vídeo), seria possível determinar a probabilidade de um conhecimento ser relevante ou não (YAN; DING, 2011). Uma abordagem similar também é utilizada pelo algoritmo de buscas do Google (PAGE et al., 1999).

1.2 Objetivos

Nesta seção são apresentados os objetivos gerais e específicos desta pesquisa.

1.2.1 Objetivo Geral

O principal objetivo desta pesquisa é desenvolver um método para a extração de conhecimento a partir de transcrições de áudio de vídeos, representando-os por meio de grafos de conhecimento.

1.2.2 Objetivos Específicos

- Construir uma base de dados de transcrições de vídeos do Youtube;
- Extrair os conhecimentos presentes nas transcrições a partir de um extrator de informações;
- Representar os conhecimentos extraídos por meio de um grafo de conhecimento;
- Desenvolver um método de refinamento e melhorias no grafo de conhecimento gerado;
- Avaliar o método proposto.

1.3 Hipóteses de Pesquisa

As hipóteses de pesquisa são:

- É possível utilizar grafos de conhecimento para representar o conhecimento extraído de transcrições de áudio;
- É possível refinar o grafo gerado para inferir novos conhecimentos.

1.4 Contribuições Científicas

As principais contribuições científicas deste estudo podem ser resumidas em: (i) contribuições científicas, (ii) contribuições aplicativas e (iii) contribuições tecnológicas.

Entende-se como contribuições científicas:

- O desenvolvimento de um método de aquisição de conhecimentos, a partir de transcrições de áudio extraídos de vídeos de redes sociais, representados sob a forma de um grafo de conhecimento;
- O desenvolvimento de um método que possibilite ganho semântico (refinamento) a partir de um conjunto de tuplas de conhecimento.

Já como contribuições aplicativas entende-se a aplicação dos métodos desenvolvidos que podem corroborar em estudos de diferentes áreas de conhecimento, além do campo da ciência da computação. São elas:

- O desenvolvimento de um *framework open source* que permite a automação de um *pipeline* de PLN;
- O desenvolvimento de um *framework* para auxiliar na análise e tomada de decisão, no âmbito de marketing digital;
- O desenvolvimento de um modelo conceitual com dados empíricos para determinar como diferentes elementos de um vídeo (como por exemplo, estilo linguístico, categoria e duração) impactam na popularidade desta publicação;
- O desenvolvimento de um modelo para identificar diferentes tipos de influenciadores digitais com base nas características do conteúdo dos canais, engajamento digital e traços pessoais dos influenciadores;

- O desenvolvimento de um método para detecção automática de conteúdo patrocinado em vídeos do Youtube.

Finalmente, entende-se como contribuições tecnológicas todo ferramental desenvolvido para os fins desta pesquisa. São eles:

- Algoritmos e códigos-fonte desenvolvidos;
- Bases de dados;

Todo o código-fonte implementado está disponível para livre acesso no repositório do projeto[5].

Em relação as bases de dados, foram desenvolvidas duas bases distintas, sendo a primeira composta por cerca de 11 mil vídeos em inglês americano (US-EN), a partir de suas transcrições de áudio e demais metadados. Já a segunda base conta com mais de 38 mil vídeos em português brasileiro (PT-BR), também a partir de transcrições de áudio e seus metadados. Ambas as bases estão disponíveis para acesso mediante solicitação no site do projeto[6].

1.5 Escopo do Trabalho

O escopo desta pesquisa limita-se ao desenvolvimento de uma abordagem de aquisição de conhecimentos, com base em transcrições de áudio oriundos do Youtube, a partir de produtores de conteúdo considerados influenciadores digitais em uma dada área ou tema. Entende-se como um influenciador digital a pessoa que além de consumir conteúdos a partir de redes sociais, o produz de forma contínua e sistemática (KARHAWI et al., 2017). Além disso, seus conteúdos devem possuir uma grande relevância em um público-alvo que legitimam suas publicações como verdadeiras, sendo considerados referências em um tipo de conteúdo (KARHAWI et al., 2017).

[5] <https://github.com/jpsanr/YouGraph-Public>
[6] <https://www.ppgia.pucpr.br/~paraiso/Projects/YouGraph/>

Capítulo 2

Fundamentação Teórica

Neste capítulo serão descritas os principais conceitos teóricos necessários para o completo entendimento deste trabalho.

2.1 Processamento de Linguagem Natural

O processamento de linguagem natural (PLN) é uma área multidisciplinar que une, dentre outros, a Linguística com a Ciência da Computação, tendo como objetivo permitir que computadores possam interpretar os significados das palavras escritas nos mais diversos idiomas (linguagens) falados pelos seres humanos. Em outras palavras as técnicas de PLN se propõem a permitir que sistemas computacionais possam analisar e realizar inferências a partir de dados não estruturados oriundos de textos escritos e da fala humana (JURAFSKY, 2000).

Diversas aplicações podem se beneficiar destas técnicas, dentre as quais podemos citar, motores de busca, sistemas de diálogo (*chatbots*), ferramentas de sumarização e indexação de textos, tradução automática, entre outros.

O PLN é uma área de pesquisa cujos primeiros trabalhos datam dos anos 1950, onde sua maior motivação era no desenvolvimento de sistemas de tradução automática (HUTCHINS, 2004). Entretanto, apesar de sua longa história, foi somente nos últimos anos que houve um grande aumento do interesse do tema pela indústria e comunidade científica, principalmente pelo advento de novas técnicas de aprendizagem de máquina e inteligência artificial (JURAFSKY, 2000).

Nas seções a seguir, serão descritas algumas tarefas de PLN essenciais para esta pesquisa.

2.1.1 Operações Básicas sobre Texto

O pré-processamento dos dados é uma atividade comum na resolução de problemas com o uso de aprendizagem de máquina.

Em processamento de linguagem natural este conjunto de procedimentos tem o nome de *operações básicas sobre texto*. Estas operações podem variar em função do objetivo final do projeto, entretanto as operações mais comuns costumam ser: *tokenização*, *filtering*, *redução de dimensionalidade das palavras* e *encoding* (obtenção de uma representação vetorial dos dados de entrada).

2.1.1.1 Tokenização

Para tornar possível o processamento de uma fonte textual, muitas tarefas do PLN requerem a separação do texto em palavras, pontuações e outros símbolos. Sem este processo, a fonte de dados configura uma longa sequência de caracteres sem significado algum para o computador (WEBSTER; KIT, 1992). Após esta separação, cada token (que pode ser uma palavra ou pontuação) é armazenado em uma estrutura de dados do tipo lista.

Em uma rápida análise podemos pensar que basta separarmos o texto pelo carácter de "espaço" para obtermos os tokens, entretanto ao realizarmos a tokenização é importante observarmos a presença de contrações no texto, além do próprio idioma que pode mudar a forma da tokenização. Na Figura 2.1 é possível observar como é feita a tokenização de uma sentença em Inglês. Nesta figura, podemos visualizar que a primeira etapa realizada é a separação das palavras pelo carácter de espaço. Após isto, são identificados todos os prefixos que compõe as sub-strings das palavras. No exemplo da imagem, é separado o carácter " (aspas) do token *" "Let's"*. Em seguida o tokenizador identifica uma exceção no token *" "Let's"*. Em outras palavras, podemos entender as exceções como um conjunto de regras criadas manualmente para tratar alguma especificidade do idioma. No caso do exemplo, foi separado a palavra "Let"do "'s"(is). A seguir o próximo token que necessita atenção é o *" N.Y!""*. Nele é necessário separar dois sufixos, sendo o carácter " (aspas) e a pontuação "!". Finalmente, neste termo também é identificado uma exceção, onde as pontuações de "N.Y"devem ser tratadas como uma abreviação e não como finais de frase.

2.1.1.2 Filtering

O filtering tem como objetivo remover palavras que possuem pouco significado semântico ou relevância em um documento. Uma das formas mais comuns de filtering é a

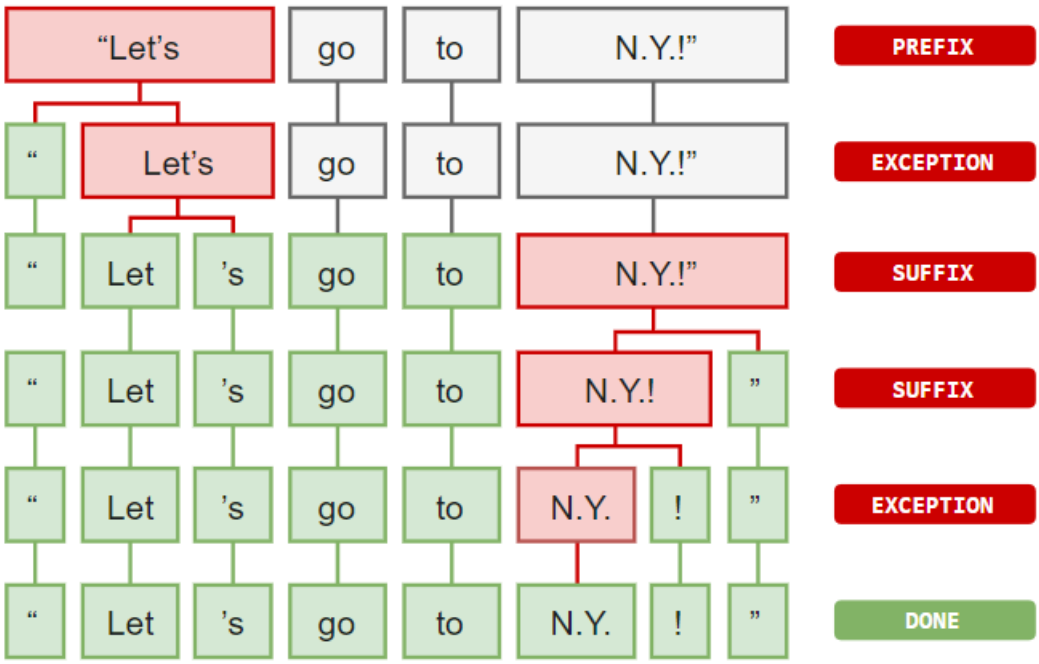


Figura 2.1: Processo de Tokenização Fonte: Spacy - <https://spacy.io/usage/spacy-101>

remoção de stop-words de um corpus. As stop-words, segundo os autores em (WILBUR; SIROTKIN, 1992), são palavras que não possuem nenhuma representatividade semântica nos significados de um texto. A baixa representatividade destes termos em um documento, se justifica em função da alta probabilidade que estas palavras têm de aparecer em diversas instâncias do corpus analisado. Palavras, como por exemplo, "eu", "está", "era", "têm"são consideradas stop-words. Existem diversas listas disponíveis, com algumas diferenças mínimas entre si. De todo modo, é importante ressaltar que em geral estas listas se concentram na remoção de artigos, pronomes e preposições. Pode-se citar o *Snowball*[1] como exemplo destas listas em português brasileiro.

Outra técnica de filtering amplamente utilizada é o algoritmo TF-IDF (do inglês term frequency - inverse document frequency) (SALTON; BUCKLEY, 1988). Ela tem como objetivo detectar a relevância de determinadas palavras em um documento presente em um corpus, levando em conta duas premissas (SEBASTIANI, 2002):

- Quanto maior a frequência que um token aparece em um determinado documento, **maior** a sua representatividade neste documento;
- Quanto maior a frequência que um token aparece no corpus, **menor** a sua representatividade em cada documento;

O valor TF-IDF de cada termo i no documento j é calculado de acordo com a

[1] http://snowball.tartarus.org/algorithms/portuguese/stop.txt

fórmula 2.1, onde $tf_{i,j}$ é o número de ocorrências de i em j, df_i é o número de documentos contendo i e N, o número total de documentos.

$$w_{i,j} = tf_{i,j} \times \log(\frac{N}{df_i}) \tag{2.1}$$

Por fim, é importante destacar que, independentemente da técnica de filtering utilizada, todas têm como objetivo diminuir a complexidade dos modelos de PLN gerados.

2.1.1.3 Normalização de Palavras

O processo de normalização de palavras possibilita a diminuição da dimensionalidade de um corpus transformando uma palavra em sua forma raiz ou similar normalizado, a fim de obter o mesmo significado da palavra original. Por exemplo, os verbos *cantar*, *cantaram*, e *cantou* podem ser representados apenas por sua forma infinitiva *cantar* (TOMAN; TESAR; JEZEK, 2006).

As duas principais estratégias de normalização de palavras são conhecidas como *lematização* e *stemming*.

O funcionamento do *stemming* é baseado na utilização de regras e heurísticas com o objetivo de encontrar a forma infinitiva e inflexionável dos termos, removendo os sufixos e prefixos (CONRADO, 2009). Exemplos de algoritmos considerados estado da arte no processo de *stemming* são os algoritmos Porter (PORTER, 1997), Stemmer (HARMAN, 1991) e Lovins (LOVINS, 1968).

Segundo os autores em (CONRADO, 2009), erros de transformação podem ocorrer no uso de *stemmers*. Eles são conhecidos como **overstemming**, quando removemos uma parte do radical da palavra, e **understemming**, quando não removemos completamente o sufixo/prefixo do termo analisado.

Em função dos erros gerados com os *stemmers*, foram desenvolvidos algoritmos de lematização. Neste caso, o processo de transformação busca encontrar o *lema* de uma palavra. Entende-se como *lema* a forma canônica de uma palavra, ou seja, a forma mais reduzida de um termo (BIRD; KLEIN; LOPER, 2009)[2]. Para isto, é necessário encontrar o *part of speech* da palavra através de um algoritmo de classificação. Após esta etapa, conhecendo a classe gramatical do termo, é possível obter o *lema* de um dado token.

Em (MANNING; RAGHAVAN; SCHÜTZE, 2008), os autores explicam que a escolha entre o *stemmer* ou o lematizador depende do propósito da aplicação a ser desenvolvida. Caso a prioridade seja a velocidade de execução, os *stemmers* tendem a ter melhores

[2]Uma forma mais didática de entender o que é um lema, é imaginar nas palavras que compõe um dicionário (BIRD; KLEIN; LOPER, 2009).

resultados. Por outro lado, caso a prioridade seja a precisão da normalização e principalmente se os resultados necessitarem da interpretação de um humano, a lematização é recomendada. Por fim, é importante ressaltar que nem todas aplicações se beneficiarão dos resultados da normalização das palavras. Por exemplo, em (HUTTO; GILBERT, 2014) é desenvolvido um modelo de análise de sentimento onde a flexão verbal altera a valência da palavra.

2.1.1.4 Encoding - Representação Vetorial

Segundo os autores em (WITTEN; FRANK; HALL, 2005), na resolução de problemas utilizando aprendizagem de máquina, normalmente tratamos strings como atributos nominais. Todavia, este artifício não pode ser utilizado quando trabalhamos com PLN, pois, desta forma, não é possível capturar e interpretar as estruturas internas presentes nos textos e seus significados.

Fica evidente a necessidade de encontrar uma estrutura que possa representar melhor estes dados textuais.

Existem diversas estruturas para este fim, como por exemplo *Bag of Words (BoW)* e os mais atuais *Word Embeddings*.

A mais simples das representações de *BoW* é o *One Hot Encoding*. Neste tipo de implementação, cada palavra no corpus é tratada como uma característica nominal e cada documento é tratado como uma instância da base. Desta forma, a presença de um termo em um documento é tratado com o valor 1 e a sua ausência com o valor 0. A Figura 2.2 ilustra um corpus de exemplo e a Figura 2.3 ilustra a representação vetorial resultante da aplicação do *one hot encoding*.

Doc 1 = "João e Maria moram juntos, Maria gosta de João."
Doc 2 = "João tem um cachorro legal"
Doc 3 = "Maria é uma garota legal"

Figura 2.2: Exemplo de corpus com 3 documentos

Bag of Words - One Hot Encoding										
	João	Maria	Moram	Juntos	Gosta	Cachorro	Garota	Legal	tem	um
Doc1	1	1	1	1	1	0	0	0	0	0
Doc2	1	0	0	0	0	1	0	1	1	1
Doc3	0	1	0	0	0	0	1	1	0	1

Figura 2.3: Representação do One Hot Encoding

Em função das limitações do *one hot enconding*, como por exemplo a alta dimensionalidade dos vetores resultantes e esparsidade dos dados, seu uso é restrito ao pré-processamento de outros algoritmos de *Word Embeddings* (TURIAN; RATINOV; BENGIO, 2010).

Outra técnica de BoW é realizar a contagem do número de vezes que determinado token aparece em um documento. Esta abordagem possui a vantagem de determinar a relevância de uma palavra neste documento. Na Figura 2.4 temos a visualização desta representação.

Bag of Words - Contagem de Frequência										
	João	Maria	Moram	Juntos	Gosta	Cachorro	Garota	Legal	tem	um
Doc1	2	2	1	1	1	0	0	0	0	0
Doc2	1	0	0	0	0	1	0	1	1	1
Doc3	0	1	0	0	0	0	1	1	0	1

Figura 2.4: Representação com contagem de frequência

Por fim, uma representação mais interessante dos BoW é montar os seus vetores calculando o TF-IDF de cada termo nos documentos. Esta técnica já foi amplamente utilizada na literatura, como por exemplo no trabalho de Arroyo-Fernández et al. (2019) e Trstenjak, Mikac e Donko (2014). Os vetores gerados são ilustrados na Figura 2.5.

Bag of Words - TF-IDF										
	João	Maria	Moram	Juntos	Gosta	Cachorro	Garota	Legal	tem	um
Doc1	0.55	0.55	0.36	0.36	0.36	0	0	0	0	0
Doc2	0.39	0	0	0	0	0.51	0	0.39	0.51	0.39
Doc3	0	0.45	0	0	0	0	0.6	0.45	0	0

Figura 2.5: Representação com contagem de frequência

Os maiores problemas do uso do BoW são a alta dimensionalidade dos dados (pois cada palavra se torna uma dimensão da matriz), alta esparsidade, e principalmente o fato de que a ordem das palavras é desprezada na geração do vetor. Outra desvantagem ao trabalhar com BoW é o fato do vetor de características gerado pela base ser totalmente aderente aos dados de entrada. Em outras palavras isto significa que o modelo gerado a partir do Bow é incapaz de processar palavras que não estão presentes no corpus de origem.

Com o intuito de resolver estes problemas, uma nova família de algoritmos de representação de características foi desenvolvida. Eles são chamados de *Word Embeddings (WE)*. O objetivo destes algoritmos é permitir a geração de um vetor de características de tamanho pré-definido (chamado de VSM – *vector space models*, independentemente

do número de termos presentes na base (TURIAN; RATINOV; BENGIO, 2010). Como cada dimensão do vetor gerado no *word embedding* não é representado por uma palavra, mas sim por diversas características (como o contexto, por exemplo), é possível realizar operações matemáticas não somente entre textos e sentenças, mas também entre palavras isoladas (CAMACHO-COLLADOS; PILEHVAR, 2018).

Os WEs podem ser classificados em dois grandes grupos: *word embeddings de palavras* e *word embeddings contextuais*.

Os *word embeddings de palavras* visam representar diferentes tokens a partir de um vetor denso de atributos. Uma das implementações mais relevantes dos WEs de palavras é o Word2Vec apresentado por Mikolov et al. (2013b). Neste trabalho, é realizado um mapeamento de todos as palavras presentes em um corpus de treinamento a partir de uma rede neural com uma camada intermediaria. Esta rede pode ser treinada de duas maneiras diferentes: através de uma arquitetura *Continuous Bag of Words (CBOW)* ou *Skip-Gram*, exemplificadas na Figura 2.6. Os dois modelos são bastante similares, a diferença é que o CBOW prevê palavras-alvo a partir das palavras de contexto de origem w, enquanto o Skip-Gram faz o inverso e prediz palavras de contexto de origem a partir das palavras-alvo $w(t)$.

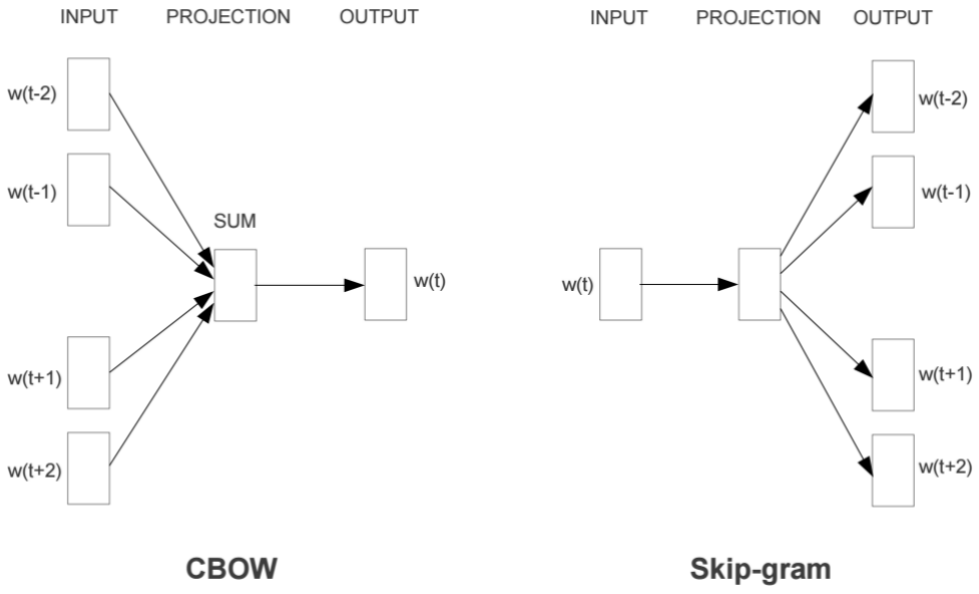


Figura 2.6: Arquitetura CBOW vs Skip-Gram. Fonte: (MIKOLOV et al., 2013b)

Por fim, além do Word2Vec, outras implementações que também merecem destaque são o GloVe (*Global Vector*) (PENNINGTON; SOCHER; MANNING, 2014) e o FastText (BOJANOWSKI et al., 2016).

Apesar dos avanços proporcionados pelos WEs de palavras, eles ainda eram in-

capazes de compreender contextos diferentes e ambiguidades (polissemia) (PETERS et al., 2018a). Para tratar o problema da polissemia foram desenvolvidos os modelos de representação dinâmicas ou também chamados de *word embeddings contextuais*. Entre as implementações dos WEs contextuais, podemos citar o ELMo (PETERS et al., 2018b), UMLFit (HOWARD; RUDER, 2018), BERT (DEVLIN et al., 2018a) e o XLNet (YANG et al., 2019).

2.1.2 Extração de Informações

Em NLP o processo de encontrar dados estruturados a partir de fontes textuais é chamado de extração de informação (em inglês: information extraction ou somente IE). Esta estruturação de dados permite encontrar entidades e atributos, bem como, os relacionamentos entre eles (chamado de relação semântica) (JURAFSKY, 2000).

Para melhor entendermos o conceito das relações semânticas, um exemplo:

João tem um cachorro chamado Mike e mora com Maria.

Desta sentença é possível inferir algumas relações semânticas, como por exemplo, afirmar que João mora com Maria. Estas relações são comumente armazenadas no formato **RDF (Resource Description Framework)** (LASSILA; SWICK et al., 1998).

RDF é uma metalinguagem que tem como objetivo formalizar a representação de informações na internet através de um identificador único. Um maior detalhamento do funcionamento do RDF pode ser obtido em (DECKER; MITRA; MELNIK, 2000). Cada instância extraída de um texto é representada no formato de tuplas com a seguinte estrutura:

tupla = (sujeito, predicado, objeto)

Desta forma, podemos inferir as seguintes tuplas no exemplo proposto:

(João, mora_com, Maria);

(João, tem_um, Cachorro);

(Cachorro, se_chama, Mike);

Segundo (JURAFSKY, 2000), os extratores de informação podem ser classificados em quatro grandes classes sendo eles:

- **Modelo baseado em regras**: Forma de extração que utiliza padrões previamente elaborados por especialistas com o objetivo de encontrá-los no texto e assim extrair as informações. Em (HEARST, 1992), estas regras foram criadas para detectar hipônimos e hiperônimos;

- **Modelo Supervisionado**: Basicamente transforma a tarefa da extração da informação em um problema de classificação. Deste modo, um anotador humano rotula uma base de treinamento com todas as tuplas detectadas por ele. Então, após este processo, a base rotulada é utilizada para treinar um modelo de classificação, para que este possa ser utilizado na classificação de novas instâncias e consequentemente extrair tuplas de textos desconhecidos. Diversos algoritmos de classificação podem ser utilizados, como por exemplo SVMs[3] (ZELENKO; AONE; RICHARDELLA, 2003), redes LSTM[4] (MIWA; BANSAL, 2016), entre outros;

- **Modelo Semi-Supervisionado (através de Bootstraping ou Supervisão distante)**: Nesta abordagem, utiliza-se de uma base rotulada como semente (bootstrap) para permitir o aprendizado de novos padrões (generalizações). Estas novas tuplas aprendidas podem ser usadas para classificar novos textos desconhecidos;

- **Modelo Não Supervisionado**: O objetivo do modelo não supervisionado é permitir a extração de tuplas sem a utilização de nenhuma base rotulada e desta forma dispensando a necessidade de um anotador humano. Para isto, é necessário montar a árvore de dependência sintática de cada sentença no texto, com objetivo de encontrar os verbos e substantivos e suas relações (part of speech). Objetivando melhorar a performance do algoritmo, bem como a qualidade da tuplas geradas, aplica-se a poda de relações muito longas, além de relações pouco frequentes;

Já em (XAVIER et al., 2014), os autores separam os extratores de informações em somente duas categorias:

- **IEs tradicionais**: são sistemas que dependem de um especialista humano para realizarem a sua modelagem e/ou utilizam algoritmos de aprendizagem de máquina para a criação de suas regras. Normalmente são dependentes de um domínio específico, uma vez que, para o sistema detectar novas tuplas é necessário gerar um novo modelo;

[3] SVM:(BOSER; GUYON; VAPNIK, 1992)
[4] LSTM: (HOCHREITER; SCHMIDHUBER, 1997)

- **IEs Abertos (Open Info Extrations)**: já na abordagem aberta, o sistema é "virtualmente"capaz de detectar infinitas tuplas de uma fonte de dados, uma vez que o seu funcionamento é feito através da análise das árvores sintáticas de sentenças presentes no texto. Desta forma, seu uso possui muito mais escala de operação;

É importante ressaltar que as definições de (JURAFSKY, 2000) e (XAVIER et al., 2014) são complementares, dado que (JURAFSKY, 2000) foca na técnica utilizada enquanto (XAVIER et al., 2014) na aplicação.

De modo geral, a escolha de qual extrator de informação utilizar varia (entre os tradicionais e abertos), pois ambos possuem vantagens e desvantagens. Nos extratores tradicionais obtém-se um formalismo maior (tornando possível o seu uso em ontologias, por exemplo). Este maior formalismo matemático é possível, pois o modelo gerado é conhecido *a priori*. Em contrapartida, a necessidade de um especialista humano para gerar o modelo torna seu uso mais custoso. Além do mais, mesmo utilizando alguma técnica de aprendizagem de máquina para treinar os modelos, sua escala continua limitada em função ao domínio restrito dos textos de entrada.

Por outro lado, os Open IEs possuem como principal vantagem a independência de domínio, sendo amplamente utilizados para a extração de tuplas de textos da Internet. No entanto, a falta do formalismo leva a uma maior dificuldade na avaliação das tuplas geradas, além da possibilidade de geração de tuplas incorretas (conceito conhecido como **semantic-drift**) (JURAFSKY, 2000).

2.1.3 Topic Modeling

Topic Modeling é o campo de mineração de texto e recuperação da informação com o objetivo de extrair tópicos latentes de um corpus. Entende-se como um tópico latente (ou escondido), um conjunto de termos ou palavras que permita caracterizar um conjunto de documentos como semelhantes ao mesmo passo que possibilite distingui-los dos demais documentos de um corpus. Desta forma, é possível classificar textos e documentos de maneira não supervisionada, para encontrar assuntos ou domínios semelhantes entre documentos diferentes (LIU et al., 2016).

Apesar de parecer uma tarefa “fácil” de ser realizada por humanos, esta tarefa é complexa em termos computacionais, uma vez que o computador desconhece os significados dos termos ou palavras em tempo de execução. Uma analogia para compreendermos o tamanho deste desafio, seria pedir para uma pessoa determinar os assuntos tratados em um conjunto de documentos em um idioma desconhecido.

Os autores em (BLEI; NG; JORDAN, 2003) explicam o funcionamento da mode-

lagem de tópicos afirmando que, para um dado corpus, existem um conjunto de palavras-chave capaz descrever diferentes assuntos (tópicos). Estas palavras-chaves são extraídas dos documentos de entrada. Cada documento do corpus é composto por um conjunto de diferentes tópicos. Desta forma, o objetivo da modelagem de tópicos é encontrar os conjuntos de palavras que representam os tópicos para finalmente determinar os tópicos dos documentos. Por exemplo no Quadro 2.7, podemos inferir que aquele trecho de uma notícia pode tratar sobre política e sobre esportes (olimpíada). Fica evidente também que algumas palavras representam mais esses tópicos que outras, como por exemplo "Olimpíadas no Rio" está para "esportes" assim como "Dilma Rousseff" está para "política". Desta forma o objetivo do topic modeling é relacionar estas palavras-chave entre si (encontrar os tópicos) para, com isso, relacionar diferentes documentos.

> BRASÍLIA - Um dia depois de o presidente interino Michel Temer visitar o Parque Olímpico no Rio, a presidente afastada Dilma Rousseff falou a internautas sobre as Olimpíadas como se ainda estivesse no Planalto. Ao lado do ex-ministro interino do Esporte Ricardo Leyser, Dilma declarou nesta quarta-feira que "mobilizará" agentes e "adotará" medidas contra o terrorismo. A petista foi impedida de continuar na Presidência após 55 senadores votarem pelo seu afastamento, no último dia 12 de maio.

Figura 2.7: Noticia veiculada no G1. Fonte: https://bityli.com/v1rZE

Existem diversos algoritmos com este propósito disponíveis na literatura, incluindo o Latent Semantic Analysis (LSA) (LANDAUER et al., 2013), Non-negative Matrix Factorization (NMF) (STEVENS et al., 2012), Probabilistic Latent Semantic Analysis (PLSA) (HOFMANN, 1999), Latent Dirichlet Allocation (LDA) (BLEI; NG; JORDAN, 2003) e o mais recente LDA2vec (MOODY, 2016), que consiste em uma combinação do tradicional LDA juntamente com o uso de word-embeddings.

Além do seu uso mais tradicional que é a classificação de documentos, as técnicas de topic modeling também podem ser utilizadas para realizar a anotação automática de imagens, e mais recentemente a classificação de genomas (ROSA et al., 2015).

2.1.3.1 Latent Dirichlet Allocation

De acordo com os autores em (BLEI; NG; JORDAN, 2003), LDA é um algoritmo generativo probabilístico, para coleções de dados discretos (como um corpora por exemplo). Além disso, o LDA é um modelo Baysiano hierárquico de três níveis, onde cada coleção é modelada através de um conjunto finito de tópicos, que por sua vez, possui

um conjunto de probabilidades. Em outras palavras, o que o LDA se propõe a fazer é: encontrar um modelo que possua uma alta probabilidade de identificar um determinado documento de um corpus ao mesmo tempo que possua uma alta probabilidade de determinar documentos similares entre si. A figura 2.8 demostra uma visualização gráfica deste modelo.

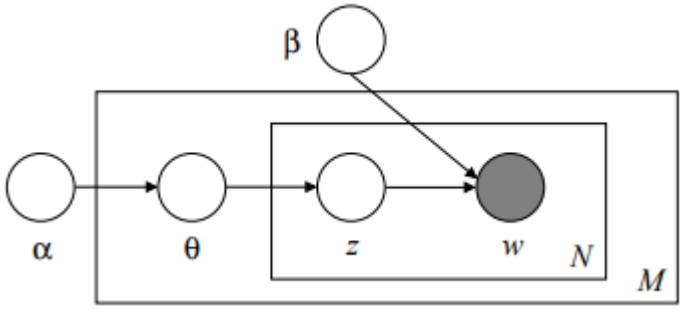


Figura 2.8: Framework do LDA. Fonte: (BLEI; NG; JORDAN, 2003)

Na figura 2.8, os parâmetros α e β são distribuições dirichlet que representam o corpus de entrada. Já a variável θ representa os documentos a serem processados (um por vez no algoritmo). Por fim, Z e ω representam as distribuições multinomiais de cada palavra em cada documento presente no corpus.

Desta forma, primeiramente o modelo define K tópicos (passado como parâmetro), onde cada k tópico é associado a uma distribuição ψ_k sobre as palavras presentes no vocabulário do corpus. Esta distribuição ψ_k, é obtida através da distribuição de Dirichlet β. De posse dos tópicos criados, em cada documento d (da coleção de palavras w_d) é gerada a sua distribuição θ_d nos K tópicos através da distribuição $_k(\alpha)$. Esta distribuição é responsável por atribuir os pesos de cada tópico em um dado documento. A seguir, de posse dos pesos dos tópicos em todos os documentos do corpus, é possível determinar o peso de cada palavra w_{di} em cada tópico presente no conjunto w_d através de uma distribuição θ_d. Por fim o LDA seleciona os tópicos $z_{di} \in [1, K]$ através de uma distribuição multinomial $_k(\theta_d)$. Finalmente cada palavra w_{di} é selecionada através da distribuição $_v(\psi_{z_{di}})$.

Contudo, apesar do LDA ser um algoritmo robusto e eficiente na obtenção dos tópicos latentes (BLEI; NG; JORDAN, 2003; MISRA et al., 2009), ele possui como limitação a exigência da definição do número de tópicos a ser encontrado em um corpus antes da sua execução. Infelizmente esta limitação pode impedir a utilização deste algoritmo em diversas aplicações, como por exemplo em corpus de domínio aberto, caso nenhuma abordagem para a inferência deste valor seja empregada.

2.1.3.2 Avaliação de Tópicos Latentes

Podemos analisar os modelos de topic modeling majoritariamente das seguintes maneiras (STEVENS et al., 2012; BLEI; NG; JORDAN, 2003):

- **Análise humana;**
- **Perplexidade;**
- **Coerência;**

O processo da análise humana consiste em um especialista avaliar os tópicos gerados e determinar se eles fazem ou não sentido no domínio dos textos.

A métrica de perplexidade avalia o quão "surpreso"o modelo fica ao receber um documento desconhecido. Isto é feito ao se analisar a verossimilhança logarítmica do conjunto de testes e treinamento através do teste de **held-out** (BLEI; NG; JORDAN, 2003). A fórmula 2.2 representa o cálculo da perplexidade. Ainda segundo Blei, por convenção da área, a função de perplexidade é **monótona não-crescente** à verossimilhança do conjunto de testes D_{teste}. Por fim, quanto menor o valor de perplexidade obtido, melhor é a generalização do modelo.

$$perplexidade(D_{teste}) = exp\left\{-\frac{\sum_{d=1}^{M} \log p(w_d)}{\sum_{d=1}^{M} N_d}\right\} \tag{2.2}$$

Finalmente a coerência permite medir a similaridade semântica entre os termos. De acordo com (HAGEN, 2018), modelos gerados com uma alta coerência tendem a gerar tópicos com uma maior interpretabilidade pelos leitores humanos. Assim, é recomendada a utilização de modelos com uma alta coerência. O cálculo da coerência pode ser realizado por meio de uma função de pontuação (ou *score*) de similaridade semântica entre termos no formato $\sum_{i<j} score(w_i, w_j)$ (O'CALLAGHAN et al., 2015). Esta função de *score* pode ser calculada de diversas formas dependendo do domínio do problema ao qual o modelo é aplicado. Dentre as medidas existentes na literatura, destaca-se a medida de associação NPMI (*Normalized Pointwise Mutual Information*) (SYED; SPRUIT, 2017). A medida de *score* NPMI é descrita a seguir (equação 2.3). Desta forma o *score* baseada na função $NPMI$ visa calcular a probabilidade logarítmica da co-ocorrência de dois termos w_j e w_i somados a uma constante ϵ.

$$score = C_{NPMI} = \frac{1}{{}^{N}C_2} \sum_{j=2}^{N} \sum_{i=1}^{j-1} \left(\frac{log\left(\frac{P(w_j, w_i) + \epsilon}{P(w_j)\ P(w_i)}\right)}{-log(P(w_j,\ w_i)) + \epsilon} \right) \tag{2.3}$$

2.2 Teoria dos Grafos

Podemos entender um grafo como um conjunto de pontos, com linhas que relacionam pares destes pontos. O nome grafo (do inglês graph - gráfico), já sugere o maior benefício no emprego destas estruturas – de prover uma representação gráfica dos dados (BONDY; MURTY et al., 1976). Os primeiros estudos da teoria dos grafos datam dos anos 1700, onde, Leonhard Euler resolveu um problema matemático conhecido como "As sete pontes de Königsberg"(EULER, 1741). Neste problema, questionava-se se seria possível ou não acessar as três ilhas que compunham a cidade de Königsberg (hoje chamada de Kaliningrado - Rússia), passando somente uma vez em todas as sete pontes existentes retornando ao ponto de origem. Para resolver esta questão, Euler representou cada ilha com um ponto (chamado de nós ou vértices) e as pontes como linhas (arestas) ligando estes pontos, conforme demostrado na figura 2.9. De posse deste grafo representando somente as pontes e massas de terra, foi possível provar matematicamente que é impossível cruzar todas as pontes sem revisitar nenhumas delas. Para que tal percurso fosse possível, seria necessário que o número de pontes fosse par, ou em termos mais formais o grau de todos os nós deveria ser par.

Apesar da teoria dos grafos ser um ramo de pesquisa antigo, ela possui inúmeras aplicações em problemas e cenários atuais. Conforme explica (SKIENA, 1998), os grafos podem ser representações abstratas com o poder de descrever organizações de transporte público, interações humanas, redes de telecomunicações e entre outras.

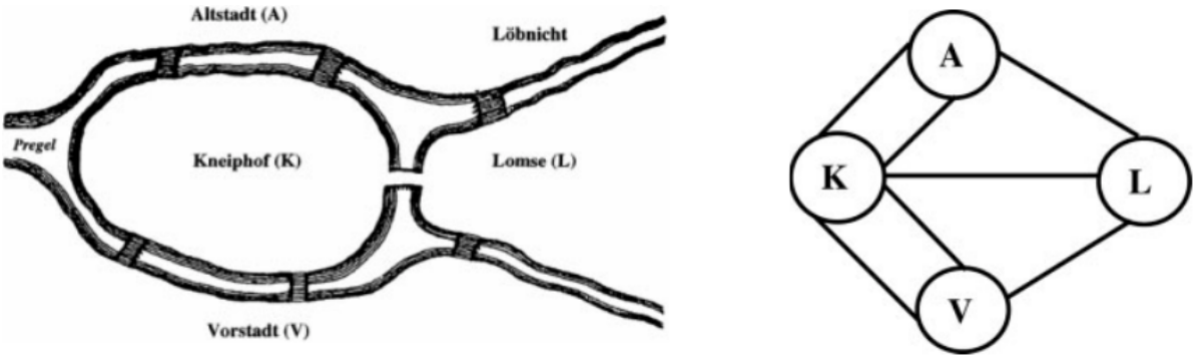


Figura 2.9: À esquerda uma ilustração de Königsberg e à direita a sua representação em grafos. Adaptado de: (GRIBKOVSKAIA; SR; LAPORTE, 2007)

2.2.1 Formalização de um Grafo

De acordo com (BONDY; MURTY et al., 1976), podemos formalizar um grafo G como sendo uma tripla ordenada $(V(G), E(G), \psi_G)$, onde $V(G)$ é um conjunto não vazio

de vértices, $E(G)$ é um conjunto disjunto de $V(G)$ de arestas e por fim uma função ψ_G que associa cada nó de G a um par de vértices. Caso este par seja ordenado $\psi_G = (x, y)$ com $\psi_G \in E(G)$, chama-se o grafo de direcionado (ou digrafo), entretanto caso este par não seja ordenado $\psi_G = \{x, y\}$ com $\psi_G \in E(G)$ (SIMÕES-PEREIRA, 2013). Na Figura 2.10 temos um exemplo de um digrafo e de um grafo não direcionado. De agora em diante neste trabalho chamaremos os vértices $V(G)$ apenas de V e as arestas $E(G)$ de E.

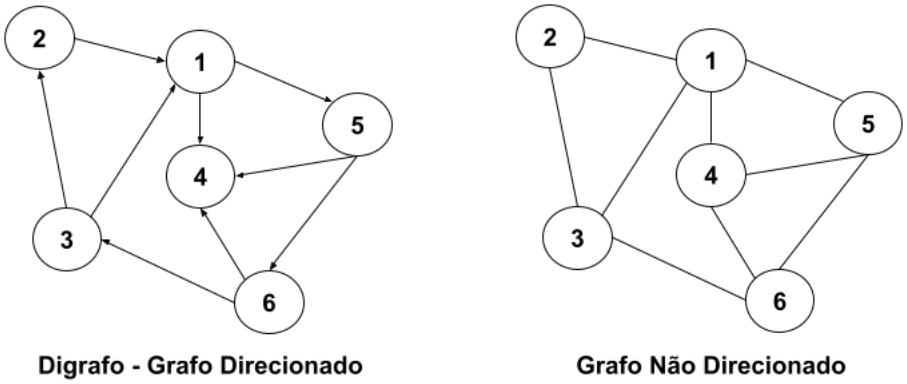


Figura 2.10: Exemplo de um digrafo (à esquerda) e de um grafo não direcionado (ou dirigido) à direita

Um dado grafo G pode ser chamado de rotulado, quando, seus nós e/ou arestas possuem algum tipo de rótulo. Um grafo rotulado é denotado por $G := (V, E, w)$, sendo V e E os conjuntos de vértices e arestas, respectivamente, e w o rótulo dos nós e/ou arestas. Os rótulos das arestas são também denominados pesos, sendo normalmente um número, ou seja, com $w : E \to \mathbb{R}$. Quando um grafo possui pesos em suas arestas ele também é chamado de ponderado (SIMÕES-PEREIRA, 2013).

Um grafo H é tido como um subgrafo de G (escrito na forma de $H \subseteq G$), quando $V(H) \subseteq V(G)$, $E(H) \subseteq E(G)$ e ψ_H é uma restrição de ψ_G em $E(H)$ (BONDY; MURTY et al., 1976).

2.2.2 Estruturas dos Grafos

Os grafos podem se diferenciar de inúmeras maneiras, como por exemplo em relação a forma da sua rede, aos atributos, à densidade e por fim aos tipos (NEEDHAM; HODLER, 2019).

2.2.2.1 Formas de um grafo

Segundo (NEEDHAM; HODLER, 2019), as redes de um grafo podem ser representadas de três formas distintas:

- **Redes Aleatórias;**
- **Redes Mundo Pequeno;**
- **Redes Scale-Free;**

Uma rede é considerada aleatória, quando um dado nó possui a mesma probabilidade p de se relacionar com qualquer outro nó da base. Desta forma esta rede não possui padrões detectáveis de relacionamentos, assim como nenhuma hierarquia e/ou comportamento de grupo. Normalmente estas redes são criadas de forma artificial com o objetivo de servir como "controle"em experimentos envolvendo testes em algoritmos para análises de grafos.

Já as redes mundo pequeno são frequentemente encontradas em bases de diversos domínios diferentes, como por exemplo em dados biológicos (PAOLA et al., 2013), em redes de transmissão de energia elétrica (WATTS; STROGATZ, 1998), redes sociais (DARAGHMI; MING, 2012) e diversos outros domínios.

A noção de mundo-pequeno ou também denominado fenômeno do mundo-pequeno surgiu num experimento proposto por (MILGRAM, 1967). Neste trabalho, foi proposto um experimento com o intuito de averiguar qual era a probabilidade de que duas pessoas nos Estados Unidos, selecionadas de forma aleatória, pudessem conhecer uma a outra. O autor concluiu que duas pessoas quaisquer que não se conhecem estão conectadas entre si através de seis pessoas intermediárias (em média). Um número que pode ser considerado relativamente pequeno se comparado com a população dos Estados Unidos (WANDERLEY, 2015). Por fim, uma rede com características de mundo-pequeno pode ser definidas como um grafo que tem as seguintes propriedades: seu coeficiente de agrupamento deve ser grande e o comprimento de caminho característico deve ser pequeno (WATTS; STROGATZ, 1998).

Já as redes scale-free, segundo (NEEDHAM; HODLER, 2019), são redes onde os graus dos nós seguem uma distribuição exponencial (CLAUSET; SHALIZI; NEWMAN, 2009). Devido ao alto grau presente neste tipo de rede alguns comportamentos começam a surgir como por exemplo: um elevado número de hubs (nós com uma alta centralidade), e nós clusterizados. O maior exemplo de uma rede scale-free é a rede mundial de computadores (World Wide Web).

2.2.2.2 Atributos de um grafo

Em relação os atributos dos grafos eles podem ser classificados das seguintes formas:

- **Conexos vs Desconexos**
- **Ponderado vs Não Ponderado**
- **Cíclico vs Acíclico**
- **Esparso vs Denso**

Antes de nos aprofundarmos na definição de conectividade de um grafo, é importante formalizarmos o conceito de *caminhos* em grafo. Deste modo, entendemos com um caminho um grafo qualquer da forma $(\{v_1, v_2, ..., v_n\}, \{v_i, v_{1+1} : 1 \leq i < n\})$. Em outras palavras, um caminho é um grafo C cujo conjunto de vértices admite uma permutação $(v_1, v_2, ..., v_n)$ tal que $(v_1v_2, v_2v_3, ..., v_{n-1}v_n = A(C)$. Se um caminho $v1...vn$ é subgrafo de G, dizemos simplesmente que $v1...vn$ é um caminho em G ou que G contém o caminho $v1...vn$. (FEOFILOFF; KOHAYAKAWA; WAKABAYASHI, 2011).

Finalmente, podemos afirmar que um grafo é conexo se, para qualquer par $\{v, w\}$ de seus vértices, existe um caminho com extremos v e w (FEOFILOFF; KOHAYAKAWA; WAKABAYASHI, 2011). Podemos citar como um exemplo de aplicações que se beneficiam do conceito de conectividade em grafos, as malhas elétricas de distribuição de energia. Desta forma podemos assumir os transformadores como vértices do grafo e as linhas de transmissão como as arrestas. Em caso de falha em alguns dos transformadores, parte da cidade ficará sem energia e esta rede ficará desconexa.

Conforme já explicado anteriormente, um grafo qualquer é ponderado quando suas arestas possuem pesos que podem ou não serem valores numéricos. Podemos citar uma aplicação de grafos ponderados as estradas rodoviárias de um país que ligam diversas cidades. Neste caso as cidades seriam os vértices, as estradas as arestas e as distancias entre as cidades (em quilômetro por exemplo) seria os pesos das arestas.

Em teoria dos grafos, entendemos como um ciclo (ou circuito) como um grafo e/ou subgrafo da forma $(\{v_1, v_2, ..., v_n\}, \{v_i, v_{1+1} : 1 \leq i < n\} \cup \{v_nv_1\})$, com $n \geq 3$. Assim sendo, um ciclo é um grafo O com $n(O) \geq 3$ cujo conjunto de vértices admite uma permutação $(v_1, v_2, ..., v_n)$ tal que $\{v_1v_2, v_2v_3, ..., v_{n-1}v_n\} \cup \{v_nv_1\} = A(O)$ (FEOFILOFF; KOHAYAKAWA; WAKABAYASHI, 2011). Um exemplo do emprego de ciclos é a detecção de comunidades em redes sociais. Pessoas próximas, se relacionam mais frequente com seus pares do que com outros elementos.

De acordo com (NEEDHAM; HODLER, 2019), a esparsidade ou densidade de um grafo, pode ser entendida como o número de relação de um grafo com o máximo de relações possíveis neste grafo. Caso todos os nós de um dado grafo se relacione com todos os outros nós deste grafo, ele é chamado de completo ou um clique. A densidade máxima de um grafo é definida na equação 2.4, onde N é o número de nós do grafo. Para medir a densidade atual de um grafo podemos utilizar a fórmula 2.5, onde R é o número de relações no grafo.

$$MaxD = \frac{N(N-1)}{2} \tag{2.4}$$

$$D = \frac{2R}{N(N-1)} \tag{2.5}$$

É importante mencionar, que inúmeros grafos baseados em redes reais tendem a ser esparsos, com aproximadamente uma correlação linear entre a sua densidade e os números de nós que o compõe (NEEDHAM; HODLER, 2019). Também é importante termos em mente que grafos muito esparsos, podem não representar as informações de forma satisfatória, ao mesmo tempo que grafos muito densos podem não agregar informações significativas ao modelo, além de aumentar a sua complexidade no processamento dos dados.

2.3 Representação do Conhecimento

O processo de representação do conhecimento visa representar os "fatos sobre o mundo"de forma computável. Desta forma, utilizando diversas técnicas como por exemplo, análise sintática e semântica e lógica preposicional, e de 1º ordem, é possível representar o conteúdo de variadas fontes de dados em uma base de conhecimento. O processo de modelagem de bases de conhecimento, bem como a formalização de conceitos oriundos de dados não estruturados, muitas vezes é chamado de *engenharia ontológica*. (RUSSELL; NORVIG, 2016). Em outras palavras a engenharia ontológica, estuda diferentes estruturas para representar conceitos e conhecimentos. Podemos entender um conceito [5] como uma generalização ou abstração de uma entidade e conhecimento como um conjunto de conceitos.

Existem diferentes estruturas que permitem o armazenamento de conceitos e conhecimentos. Entre elas, podemos citar as *ontologias* (FUNG; BODENREIDER, 2019),

[5] Podemos citar um exemplo de conceito o município de Curitiba, que pode ser representado como uma entidade do tipo cidade.

taxonomias (GANI et al., 2016), *folksonomias* (WAL, 2009), *grafos* (PAULHEIM, 2017) e *frames* (MINSKY, 1974). Todas as estruturas citadas acima são especificações de redes semânticas.

2.3.1 Redes Semânticas

Segundo (SOWA, 1987) e (LEHMAN, 1992), redes semânticas podem ser entendidas como grafos que representam estruturas de significados. Desta forma, estes significados originam representações de conceitos que interagem com outros conceitos através de relações de arestas (arcos do gráfico). Segundo (HARTLEY; BARNDEN, 1997) existem diversas vantagens em utilizar grafos para representar conceitos. Entre elas, podemos citar:

- **Visibilidade:** Os grafos são claros e objetivos para leitura humana;
- **Formalismo:** Redes semânticas (e suas relações) podem ser escritas utilizando lógica de primeira ordem e consequentemente realizar inferências de conhecimentos implícitos (LEHMAN, 1992);
- **Generalização:** Através de formas genéricas e relações é possível modelar conceitos complexos (Objetos, situações, eventos);

2.3.2 Grafos de conhecimento

Apesar do termo grafo de conhecimento (*Knowledge Graph* - GC) ter sido amplamente difundido pela Google em 2012 (SINGHAL, 2012), seu uso na comunidade acadêmica é antigo, tendo trabalhos mencionando-o desde os anos 70 (HOGAN et al., 2021). Basicamente o emprego do termo grafo de conhecimento significa permitir que grafos compreendam entidades do mundo real, bem como os seus relacionamentos entre si (SINGHAL, 2012). Infelizmente, esta definição, é subjetiva, de modo que outros pesquisadores buscam encontrar uma definição mais técnica e formal para o termo. Até o presente momento, a comunidade acadêmica ainda não chegou a uma definição unificada do tema.

Segundo a revisão bibliográfica realizada por (HOGAN et al., 2021), podemos classificar as definições dos GCs em quatro categorias principais:

- **Categoria 1:** nesta definição os autores consideram um GC, qualquer grafo que possua entidades que se relacionam com outras entidades. Estas relações normalmente ocorrem através de grafos direcionados e rotulados (LIN et al., 2015), (WANG et al.,

2014). A principal crítica a esta definição, é que não fica claro a diferença em grafo (no sentido de banco de dados) e um GC. Além disso o conceito de conhecimento também não é explorado (WANG et al., 2017).

- **Categoria 2:** define que um GC é uma base de conhecimento estruturada na forma de grafos (NICKEL et al., 2015), (SEUFERT et al., 2016). Novamente esta definição levanta questionamentos, sobre qual seria a definição de uma base de conhecimentos e principalmente levanta algumas ambiguidades, no sentido de que, uma ontologia poderia ser ou não considerado um GC.

- **Categoria 3:** preconiza que, para um grafo ser considerado um GC, os seguintes requisitos devem ser atendidos: descrever entidades e relações do mundo real; definir as classes e relações de uma entidade em um esquema formal, permitir a inter-relação arbitrária entre diferentes entidades; e finalmente cobrir diferentes tópicos de diversos domínios (PAULHEIM, 2017).

- **Categoria 4:** Bonatti em (BONATTI et al., 2019), sugere uma nova visão na definição de um GC. Ao invés de definir formalmente o que é um GC, ele propõe a apresentação de exemplos para afirmar o que seria ou não um grafo de conhecimento. Desta forma ao analisarmos a DBPedia, FreeBase, e outros, podemos compreender que GCs são estruturas que representam objetos de interesse e suas conexões (HOGAN et al., 2021).

Existem diversos GC disponíveis atualmente com distinções entre si. Suas distinções se aplicam aos seguintes critérios (PAULHEIM, 2017):

- **Ao domínio:** Domínio aberto ou específico;

- **Ao uso:** Gratuito ou pago;

- **À curadoria:** Desenvolvido por engenheiros do conhecimento ou comunidade;

Além disso, do ponto de vista de aplicações, os GCs também podem ser classificados como *Open Knowledge Graph* e *Enterprise Knowledge Graph* (HOGAN et al., 2021). Os Open Knowledge Graphs, são disponibilizados de forma online e são desenvolvidos para uso do publico em geral. Eles podem ser de domínio aberto ou fechado, e seus dados podem ter sido extraídos da Wikipedia como no caso do DBpedia (LEHMANN et al., 2015), ou desenvolvidos pela comunidade como ocorre com o Wikidata (VRANDEČIĆ; KRÖTZSCH, 2014).

Os Enterprise Knowledge Graphs, são desenvolvidos com o objetivo de servirem a um propósito específico de uma empresa, como por exemplo em motores de buscas (SINGHAL, 2012) e sistemas de recomendação (NOY et al., 2019). Desta maneira eles não são disponíveis ao público em geral.

Entre os principais GCs disponíveis atualmente podemos citar o DBpedia, Wikidata, Yago (HOFFART et al., 2011) e GeoNames [6].

Quando tratamos de grafos de conhecimento existem três grandes tarefas envolvidas: a **geração**, o **refinamento** e a **avaliação** de um GC.

2.3.2.1 Geração de Grafos de Conhecimento

Consiste na extração de informação de uma fonte de dados e a sua representação em um grafo. Cabe ressaltar que existem outros extratores de informações para outros tipos de dados (imagens, banco de dados relacionais, ...), que não entram no escopo deste trabalho.

Ao gerar um novo grafo de conhecimento, um aspecto relevante a ser definido é a escolha de qual modelo de representação de dados será utilizado (HOGAN et al., 2021). Os três modelos mais comumente utilizados são:

- **Grafos direcionados e rotulados:** é definido como um conjunto de nós e um conjunto de arestas que relacionam estes nós. Desta forma cada nó deste grafo representa uma entidade e as suas arestas representam uma relação binária com outros nós. O DBpedia é um exemplo clássico desta abordagem;

- **Grafos de datasets:** Consiste em um conjunto de grafos (ou subgrafos) nomeados com fins específicos. Desta forma é possível tratar um grafo nomeado como um único nó de outro grafo. Este conceito é especialmente interessante quando trabalhamos com linked data, dado que está técnica permite relacionar e interagir com diversos grafos de origens distintas.

- **Grafos de propriedades:** Os grafos de propriedade foram criados para fornecerem uma maior flexibilidade na modelagem dos dados. Eles permitem tratar as arestas como um objeto e desta forma adicionar novas propriedades nela, além do seu próprio rótulo. O banco de dados Neo4j é um exemplo da utilização de grafos de propriedades;

A escolha de qual modelo de grafo utilizar, automaticamente, determinará qual a linguagem de consulta será empregada na fase da inferência dos dados. Enquanto os

[6] <http://geonames.org/>

grafos direcionados e rotulados e de datasets se baseiam no protocolo RDF, sua linguagem de consulta é o *SPARQL*. Por outro lado, caso seja utilizado o modelo de propriedades, a linguagem de consulta padrão é o *Cypher* (caso seja utilizado o *Neo4j*[7]).

Um maior formalismo destas representações, bem como das linguagens de consultas empregadas, pode ser encontrado em (ANGLES et al., 2017).

2.3.2.2 Refinamentos dos Grafos de Conhecimento

Por definição, todo grafo de conhecimento é sempre considerado incompleto e com possíveis erros lógicos e semânticos em seus nós (HOGAN et al., 2021), (PAULHEIM, 2017), (PUJARA et al., 2013).

Isto acontece principalmente por três razões:

- **Fonte de dados incompleta:** a fonte de dados pode não conter todas a informações necessárias para gerar um GC completo;
- **Erros na extração de informação:** qualquer falha no processo da extração da informação irá propagar o erro para o GC. Isso ocorre por exemplo quando lidamos com termos ambíguos. A título de exemplo o termo "Getúlio Vargas", pode ser interpretado como o ex-presidente brasileiro, uma cidade, ou ainda como um nome de rua. Caso esta distinção não seja feita da forma correta, diversas relações incongruentes surgirão no grafo;
- **Fonte de dados com dados conflitantes:** Muitas vezes, a construção de grafos de conhecimentos é feita consumindo-se diversas fontes de dados diferentes (criadas por diferentes autores). Desta forma, conflitos de informações podem ocorrer nos dados de origem (PASTERNACK; ROTH, 2010);

Desta forma é de extrema importância a detecção destes erros para realizar ajustes e correções no GC. Esta tarefa possui o nome de **refinamento de GCs**.

Duas atividades estão relacionadas ao refinamento de GCs: **conclusão de grafos** e **detecção de erros**. Dentre os diversos métodos desenvolvidos para este fim, eles podem ser classificados como **externos** e **internos**. Os métodos externos, permitem o refinamento dos GCs através de fontes de dados externas (outro GC ou corpora) que não constitui o grafo de conhecimento em análise. Por outro lado, os métodos internos, permitem a utilização única e exclusivamente do próprio GC avaliado.

[7] <https://neo4j.com/>

A tarefa de conclusão de grafos tem como objetivo aumentar a cobertura dos GCs, predizendo possíveis nós e relações ausentes na base de conhecimento. Ela também permite detectar o tipo (por exemplo a label de uma entidade nomeada) do nó ausente quando grafo possui um esquema definido *a priori*. Basicamente, a conclusão de grafos é um problema de classificação de dados. No trabalho de (SLEEMAN; FININ, 2013) o autor utiliza o algorítmo SVM para predizer o tipo de nó em um GC (FreeBase [8]). Já em (SLEEMAN et al., 2015) os autores utilizam o algoritmo de topic modeling LDA para classificar o tipo do nó. Em (LANGE; BÖHM; NAUMANN, 2010) os autores trabalham *Conditional Random Fields* para predizer relação a partir dos info boxes da Wikipedia.

A detecção de erros em GC divide-se em duas categorias: **validação de fatos** e **reparação de inconsistências** (HOGAN et al., 2021).

O processo de validação de fatos procura calcular a probabilidade de uma relação de dois nós serem verdadeiras. Para isso, em cada relação atribui-se um valor chamado de plausabilidade ou score de veracidade. Existem diversas técnicas para a validação de fatos. Em (PASTERNACK; ROTH, 2010) os autores trabalham com uma variação do algoritmo HITS (KLEINBERG, 1999) para calcular a autoridade do recurso de origem (fonte de dados). Desta forma, é possível determinar a confiabilidade de uma informação a partir de sua origem. Já em (CIAMPAGLIA et al., 2015) o autor trabalha com o conceito de proximidade semântica para verificar a veracidade de um fato.

2.3.2.3 Avaliação de Grafos de Conhecimento

Como apresentado anteriormente, os grafos possuem a premissa de serem incompletos e com possíveis erros. Isto posto, é importante realizar constantes avaliações para determinar a qualidade do GC. De acordo com (IOSUP et al., 2016), o processo de avaliação de grafos não é trivial devido a diversas abordagens de implementações, além da alta variabilidade dos tipos dos dados. Desta forma, (IOSUP et al., 2016) e (HOGAN et al., 2021) sugerem algumas famílias de algoritmos para avaliação de grafos:

- **Centralidade:** Visa encontrar os nós mais relevantes em um grafo.
- **Detecção de Comunidades:** Procura encontrar sub-grafos fortemente conectados dentro de um grafo;
- **Conectividade:** Busca estimar o quão conectado está um nó do grafo em relação aos demais. Permite também, detectar nós inatingíveis (que não possuem arcos

[8] O freebase era um GC público aberto, que foi adquirido pela Google em 2010. O projeto do Freebase foi descontinuado em maio de 2016.

direcionados para ele).

- **Similaridade:** Procura encontrar nós que são similares entre sí, do ponto de vista das vizinhanças e tipos de arcos.
- **Path Finding:** Família de algoritmos para encontrar um caminho entre dois nós.

Na seção 4.4.1 são apresentados alguns exemplos práticos de uso destas métricas.

Capítulo 3

Trabalhos Relacionados

Neste capítulo apresentamos os principais trabalhos que se relacionam com esta pesquisa. Para encontrar estes trabalhos foram realizadas consultas nas seguintes bases de artigos científicos:

- **IEEE**: <https://ieeexplore.ieee.org/>
- **ACM**: <https://dl.acm.org/>
- **Scopus**: <https://www.scopus.com/>
- **Science Direct**: <https://www.sciencedirect.com/>
- **Sprinker Link**: <https://link.springer.com/>

Utilizando a string de busca presente no Quadro 3.1, foi realizado um levantamento sistemático utilizando como parâmetros de busca, artigos a partir de 2010 em conferências, periódicos e livros. A busca foi realizada no modo *full text*[1] [2] A última consulta para verificação de novos trabalhos foi realizada no dia 12 de junho de 2021.

Como critério de inclusão, foram adicionados artigos primários, secundários e terciários. Como critério de exclusão, foram removidos artigos duplicados, literatura cinza[3], patentes e reimpressões de artigos originais.

Todos os trabalhos retornados estão inseridos nas seguintes áreas de pesquisa: *Engenharia*, *Ciência da Computação* e *Marketing*.

[1]Full Text significa permitir que a string de busca pesquise por ocorrências das palavras chave no texto do artigo, titulo e abstract.

[2]Vale mencionar que foi gerada uma *string* de buscas análoga a apresentada com termos em PT-BR, entretanto nenhum trabalho relevante foi retornado

[3]Neste trabalho, entende-se como literatura cinza, artigos que não tenham sido avaliados em uma revisão por pares.

("Youtube" OR "video" OR "online video" OR "social media") AND
("feature extraction" OR "metadata" OR "unstructured data" OR "data mining" OR
"influencer marketing" OR
"product promotion" OR "Knowledge based systems" OR "big data" OR
"heterogeneous data" OR "information extraction"
OR "unstructured data")

Figura 3.1: String de busca utilizada

Os artigos encontrados foram divididos em duas categorias: (1) sistemas de extração de informação a partir de redes sociais baseadas em vídeos e (2) grafos de conhecimento como ferramentas de representação do conhecimento.

3.1 Sistemas de extração de informação a partir de redes sociais baseadas em vídeos

Nesta subseção serão apresentados os trabalhos que de alguma forma atacaram o problema da extração de conhecimentos a partir de vídeos da Internet. Os trabalhos encontrados se diferenciam em relação ao tipo do dado utilizado e ao objetivo do estudo.

Em relação às fontes de dados, basicamente quatro fontes não estruturados podem ser utilizadas: (i) *frames* das imagens, (ii) conteúdo falado – a partir de transcrições e/ou análise de áudio (*Speech analysis*) –, (iii) comentários, ou ainda (iv) a partir de metadados dos vídeos – número de visualizações, total de comentários, curtidas, *etc.*

No âmbito dos objetivos, podemos classificá-los em: (i) sistemas de recomendação, (ii) sistemas de análise de sentimentos, (iii) classificação de vídeos e finalmente, (iv) agrupamento de vídeos.

No trabalho de Davidson et al. (2010) é apresentado um algoritmo de recomendação, que tem como objetivo manter o usuário engajado no uso da plataforma sugerindo recomendações de vídeos personalizadas para cada usuário, a partir de suas interações passadas no Youtube. Para que este objetivo fosse atingido, os autores separaram este problema em duas etapas: identificação de vídeos relacionados e geração de candidatos a recomendação. Para a execução de ambas as etapas, foi empregado um grafo direcionado ponderado, onde o nó pai é um vídeo semente (vídeo que está sendo visualizado pela audiência) e os nós filhos são os vídeo recomendados.

- **Identificação de Vídeos Relacionados:** Nesta etapa é realizado um mapeamento a partir de um vídeo v_i para um conjunto de vídeos R_i relacionados. Neste contexto,

os autores assumem que: se um usuário está assistindo um vídeo no momento (vídeo semente v), ele possivelmente irá continuar assistindo vídeos similares a este a seguir. O processo para identificação de vídeos relacionados é feito a partir da mineração de regras de associação.

- **Geração de Candidatos a Recomendação:** Uma vez identificado um conjunto de vídeos relacionados, é necessário gerar uma lista de recomendações ao usuário. Para isto, juntamente com os vídeos candidatos, também são adicionados vídeos já assistidos pelos usuários (e marcados com a *tag* "gostei"), listas de reprodução (do inglês *playlist*) que compõe o vídeo assistido ou ainda vídeos favoritados. Entretanto, conforme apontado pelos autores, esta abordagem não é resiliente o suficiente para indicar ao usuário vídeos inéditos e/ou diferentes aos comumente assistidos pelo telespectador. Desta forma, vídeos muitos similares ao assistido no momento são removidos da lista para permitir a adição de novos conteúdos.

Em (COVINGTON; ADAMS; SARGIN, 2016), os autores apresentam uma evolução ao trabalho de Davidson et al. (2010), entretanto baseadas em redes de aprendizado profundo (*Deep Learning*) e *embeddings*. Desta forma, foram treinadas duas redes a partir de um corpus de milhões de vídeos e seus metadados. A primeira rede neural, é responsável por encontrar os vídeos candidatos à recomendação (*candidate generation*) e a segunda por ranquear estes candidatos.

No mapeamento sistemático realizado, a subárea que mais retornou trabalhos, foi a referente à análise de sentimentos. Entretanto, apesar dos propósitos dos trabalhos serem parecidos (analisar o sentimento a partir de dados de redes sociais baseadas em vídeos) , os métodos utilizados diferem tanto em relação as técnicas empregadas, quanto em relação as fontes de dados. Por exemplo, Kaushik, Sangwan e Hansen (2013) em sua pesquisa, elaborou um classificador binário de valência para determinar se postagens sobre *reviews* de produtos eram positivas ou negativas. Para isto, os autores primeiramente geraram um modelo acústico para extração de características sonoras da fala (a partir de um modelo oculto de Markov – *HMM*). Então, este modelo foi utilizado para converter as falas em transcrições textuais, que posteriormente foram treinadas em um modelo de máxima entropia (do inglês *maximum entropy* – *ME*) para determinar a sua valência. No processo do treinamento do modelo foram utilizados dados de avaliações de *e-commerces* como a Amazon. Já em (BHUIYAN et al., 2017), os autores constatam (de forma análoga à Davidson et al. (2010)) que determinar a relevância de um vídeo não é uma tarefa trivial. Desta forma, os autores apresentam um modelo para determinar a relevância de um dado vídeo a partir de valência dos comentários feitos pelos telespectadores desta postagem a

partir de um dicionário de léxicos. Por fim, a partir da abordagem realizada, os autores conseguiram uma acurácia de cerca de 75% em determinar se um vídeo era relevante ou não.

Em relação a classificação de vídeos, podemos citar os seguintes trabalhos: (RAMESH et al., 2020), (ZHANG et al., 2018) e (GERHARDS, 2019). Em (RAMESH et al., 2020), os autores propõem um método para classificação de vídeos em educacionais ou não educacionais. Para isto, foram utilizados os metadados dos vídeos (palavras-chave), juntamente com os seus *frames* no processo de treinamento. Para a extração de características textuais foi utilizada uma abordagem de *words embeddings* (ELMo), enquanto que para a extração de características dos *frames* foram utilizadas redes neurais convolucionais (CNN). Por fim, para a classificação dos vídeos, dado que é um problema de classificação binária foi utilizado uma rede *Multi-Layer Perceptron*.

Em relação ao agrupamento de vídeos, encontram-se trabalhos que categorizam publicações do Youtube de forma não supervisionada. Isso pode ser feito a partir de algoritmos de modelagem de tópicos, como em (STAPPEN et al., 2021), onde a partir das transcrições de áudio dos vídeos os autores mapearem os tópicos latentes, a partir de um modelo de LDA (*Latent Dirichlet Allocation*. Após análise humana, os autores utilizaram o modelo gerado para determinar vídeos que realizam *reviews* de produtos. Já em (THELWALL, 2018), é apresentado um novo *framework* (chamado de CTFC – Comment Term Frequency Comparison), que tem como objetivo categorizar e analisar os comentários realizados pelas pessoas em vídeos relacionados ao domínio de dança em geral. Um ponto que merece destaque neste trabalho, é o fato dos autores utilizarem métodos de análise de redes (grafos) com o objetivo de encontrar grupos (*clusters*) de estilos musicais correlatos e suas danças.

3.2 Grafos de Conhecimento como Ferramentas de Representação do Conhecimento

Nesta seção serão apresentados os trabalhos que abordam o processo de recuperação da informação e/ou inferência de novos conhecimentos em GC.

Em (CIAMPAGLIA et al., 2015) é apresentado uma abordagem para checagem de fatos baseado em grafos de conhecimento. Para isto, os autores desenvolveram uma nova métrica chamada de *proximidade semântica*. Esta métrica é derivada do cálculo do menor caminho entre o fecho transitivo de dois nós [4]. A intuição por trás destes cálculo é

[4]Mais detalhes sobre teoria dos grafos na seção 2.2.

que quanto maior o caminho necessário de um nó x até y maior é a probabilidade destes dois nós não possuírem uma conexão verdadeira. O maior benefício desta abordagem em relação a outros trabalhos estado-da-arte é o fato dela transformar o processo de validação de fatos em um problema de análise de redes em grafos. Os autores finalizam afirmando que apesar dos resultados animadores, muitos trabalhos futuros ainda podem ser desenvolvidos como por exemplo, testar outros algoritmos de menor caminho.

3.2.1 Sistemas de Q&A baseados em grafos de conhecimento

Em Unger, Freitas e Cimiano (2014) é discutido aspectos importantes sobre os conceitos de *linked data* e na forma em que o relacionamento de diversos grafos de conhecimento possibilitam um maior acesso a informações para os sistemas de perguntas e respostas (Q&A). Continuando o raciocínio dos autores surgem diversos desafios em trabalhar com Q&A e linked data, dentre os quais podemos citar:

- **Diversidade de datasets disponíveis:** Um dos principais problemas ao se trabalhar com diversos grafos de conhecimento é a existência de dados conflitantes. Além disso, outro problema recorrente é permitir a integração de GCs de diferentes idiomas. Em teoria, o padrão *RDF* é multi-idioma (uma vez que se baseia em identificadores únicos para cada nó), entretanto na prática a conversão dos rótulos para outro idioma não é trivial, e até o presente momento, poucos trabalhos exploram esta questão.

- **Mapear uma pergunta realizada em linguagem natural para convertê-la em uma linguagem de consulta apropriada:** Dado que o usuário interage com o sistema utilizando alguma linguagem natural, é necessário converter este texto em alguma estrutura computável ao grafo de conhecimento. Ao se utilizar o padrão RDF (protocolo comumente utilizado em GCs públicos) isto é feito utilizando a linguagem de consulta SPARQL. Entretanto, a dificuldade neste processo é principalmente no tratamento de ambiguidades da linguagem.

- **Performance e escalabilidade:** Dado que os GCs públicos normalmente possuem milhões de triplas, o processo de busca é altamente custoso computacionalmente falando. Adicionalmente a isto, milhares de usuários e sistemas consumindo estes dados geram um enorme problema de escala e performance.

Unger e seus colegas também apresentam uma representação genérica dos principais componentes envolvidos em um sistema de Q&A (vide figura 3.2).

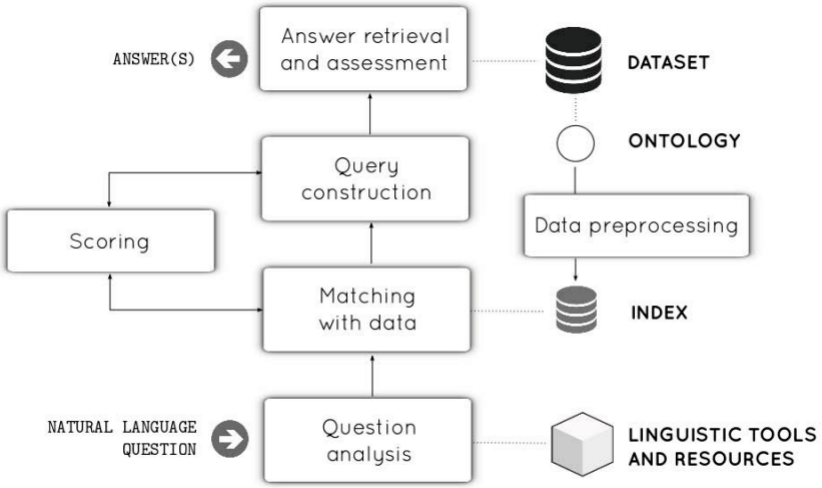


Figura 3.2: Principais componentes de um sistema de Q&A. Fonte: (UNGER; FREITAS; CIMIANO, 2014)

Como é comum em um sistema baseado em PLN, a primeira etapa realizada é o **pré-processamento** dos dados. Após, é realizada uma **análise da questão** de entrada, com o objetivo de montar um vetor de característica do texto de entrada. Este vetor depende das particularidades do sistema utilizado, mas em geral, consiste na extração de entidades nomeadas, part-of-speech (POS) e relacionamento com dicionários externos (como por exemplo o WordNet [5]). Para tratar as ambiguidades presentes no vocabulário de entrada é feito o **data matching**. Em outras palavras esta etapa é responsável por converter os termos de entrada em uma terminologia conhecida pelo modelo do sistema. Feito isso é finalmente possível construir uma consulta de buscas com a entrada do usuário (**query construction**). Normalmente após a construção da consulta é aplicada uma função de **scoring** para determinar uma pontuação, tanto da consulta gerada, bem como da entrada do usuário. Isto é importante para tratar eventuais ambiguidades, analisando as similaridades dos termos encontrados. Por fim, é apresentado uma resposta à entrada do usuário (**answer presentation**). Esta resposta pode ser apresentada na forma visual (através de um grafo) ou textualmente (através da geração de um texto com a resposta).

Os sistemas de Q&A baseado em linked data podem ser categorizados nas seguintes abordagens:

- **Abordagem baseada em controle de linguagem natural:** Utilizando léxicos [6] previamente definidos os sistema consegue restringir a dimensionalidade dos dados de entrada, de forma a garantir que a consulta de busca esteja dentro do escopo

[5] <https://wordnet.princeton.edu/>

[6] No contexto de PLN podemos entender os léxicos como um conjunto de palavras previamente selecionadas e mapeadas pelo sistema de forma a garantir a compreensão do seu sentido.

da base de conhecimento. Esta abordagem é especialmente útil em Q&As de domínio fechado. Podemos encontrar um exemplo desta abordagem em (BERNSTEIN; KAUFMANN; KAISER, 2005);

- **Baseada em gramática formal:** Entende-se como sistemas baseados em linguagem formal, ferramentas que utilizam representações sintáticas e semânticas (como árvores de dependência, por exemplo) ao analisar a entrada do usuário. O ORAKEL (CIMIANO et al., 2008) e o Pythia (UNGER; CIMIANO, 2011) são dois exemplos desta abordagem. Em ambos os trabalhos, o texto de entrada do usuário é convertido em uma representação sintática (árvore de dependência) para posteriormente, ser utilizado em um léxico e uma ontologia pré-definida, gerando uma representação semântica da entrada. Feito isso, gera-se uma consulta para buscar a resposta do usuário. Esta abordagem tem a limitação de necessitar de um léxico e consequentemente seu escopo é de domínio fechado;

- **Abordagens baseadas no mapeamento de estruturas linguísticas para estruturas semânticas compatíveis com ontologia:** Podemos citar diversos sistemas baseados nesta abordagem como por exemplo o Aqualog (LOPEZ et al., 2007), PowerAqua (LOPEZ et al., 2012), QAKIS (CABRIO et al., 2012), FREyA (DAMLJANOVIC; AGATONOVIC; CUNNINGHAM, 2011), entre outros. Seu funcionamento se baseia em utilizar o conceito de **triple query**, que extrai os principais conceitos presentes na entrada do usuário. Podemos imaginar este processo de forma análoga aos extratores de informações explicados na seção 2.1.2. Feito isso, busca-se por sinônimos, hiperônimos e hipônimos aos termos utilizando um dicionário como o WordNet por exemplo, para então buscar uma tripla RDF que resolva a entrada do usuário;

- **Abordagem baseada em template:** Seu funcionamento é similar à abordagem baseada em gramática formal, com diferença de não ser limitado à um domínio específico. Em (UNGER et al., 2012) é apresentado o termo **pseudo-query**, que funciona como uma camada intermediária entre a entrada do usuário e a consulta propriamente dita. Esta pseudo-query é gerada *parsseando* o texto de entrada retirando os part-of-speech (POS) presentes. Através de um processo de classificação é encontrada uma consulta pré-definida (utilizando heurísticas) que resolva a entrada. Por fim, após encontrar uma pseudo-consulta que corresponda aos POS encontrados, os POS são substituídos pelos termos de entrada para finalmente gerar a consulta SPARQL. A maior desvantagem desta abordagem é depender de templates ou pa-

drões previamente criados, que podem não corresponder ao texto de entrada;

- **Abordagem em exploração de Grafos:** Conforme apresentado por (FREITAS et al., 2013), (TRAN et al., 2009) e (SHEKARPOUR; NGOMO; AUER, 2013), os grafos podem ser utilizados para interpretar a linguagem natural de forma a mapear os termos presentes na entrada do usuário e encontrar entidades e relações que correspondam a elas em um grafo de conhecimento. A principal desvantagem desta abordagem é o seu alto custo computacional de processamento, que é diretamente proporcional ao tamanho da base de conhecimento utilizada. Para tentar resolver isto, (FREITAS et al., 2013) vai além, e implementa o conceito dos **graph-embeddings** para tentar reduzir a dimensionalidade dos dados e otimizar as consultas. Adicionalmente a isto, os autores ainda implementaram diversas heurísticas para facilitar a busca pelas informações.

- **Abordagens baseadas em aprendizagem de máquina:** Atualmente diversos autores propõem tratarmos sistemas de Q&A como um problema de aprendizagem de máquina. Desta forma, até mesmo as abordagens descritas anteriormente, podem se beneficiar destes algoritmos. Isto ocorre por exemplo em (FREITAS et al., 2013) que utiliza os graph-embeddings para inferir a melhor resposta ao usuário. Unger em (UNGER et al., 2012) também trabalha com classificadores durante o processo de inferência. Além disto diversas implementações, que apesar de não trabalharem especificamente com aprendizagem de máquina, usufruem destas ferramentas ao encontrar o POS e NERs nos textos de entrada (UNGER; FREITAS; CIMIANO, 2014).

3.3 Considerações Finais sobre o Estado da Arte e Lacunas de Pesquisa

Neste capítulo, foram apresentados os principais trabalhos que de alguma forma se relacionam com o escopo desta pesquisa. É importante mencionar que o mapeamento sistemático realizado teve como objetivo encontrar os principais trabalhos (estado da arte) que podem contribuir com o desenvolvimento desta dissertação, de forma que, não foi realizada uma busca extensiva por pesquisas correlatas. Também vale ressaltar que além dos trabalhos discutidos neste capítulo, foram encontrados outros trabalhos que são aderentes à esta pesquisa em menor grau. A Tabela 3.1 apresenta uma lista adicional com outros trabalhos similares.

Por fim, após o levantamento do estado da arte referente a esta dissertação, algumas lacunas de pesquisa ficaram evidentes. Praticamente todos os autores apontaram dificuldades em trabalhar com dados oriundos do Youtube. Devido a magnitude desta rede (onde os números de vídeos publicados, usuários cadastrados e comentários postados estão na ordem de bilhões), processar, analisar e explorar estes dados não é trivial. Indo além, a heterogeneidade dos dados também dificulta o seu uso, uma vez que eles podem ser textuais, visuais (imagens) ou ainda sonoros. Desta forma, alguns autores já demonstraram o interesse no desenvolvimento de ferramentas que favoreçam o consumo deste tipo de informação como em (ADNAN; AKBAR, 2019), (SHAIKH et al., 2018) e (DABBÈCHI et al., 2017), onde são apresentados *frameworks* baseados em computação distribuída que realizem o armazenamento, processamento e visualização destes documentos, mas estes trabalhos ainda são incipientes (ADNAN; AKBAR, 2019).

Uma outra lacuna de pesquisa identificada, é a necessidade de especialistas humanos na execução dos experimentos como em (GERHARDS, 2019) onde os autores dependeram da execução de um questionário para mapear os perfis dos influenciadores no Youtube. Outro caso, é em (SCHWEMMER; ZIEWIECKI, 2018), onde os autores também empregaram especialistas humanos para encontrar o melhor modelo de tópicos (LDA) que melhor representa um corpus de transcrições de áudio.

Por fim, em (THELWALL, 2018) é apresentado o uso de grafos para analisar o comportamento de redes no Youtube, mas o conceito de grafos de conhecimento não foi explorado em nenhuma pesquisa.

Título do Trabalho	**Tipo do Dado**	**Objetivo**	**Autores**
EDUCATION DATA MINING: MINING MOOCS VIDEOS USING METADATA BASED APPROACH	Video, Áudio e Metadados	Agrupamento	(ABDELALI et al., 2016)
Will Sentiments in Comments Influence Online Video Popularity?	Comentários	Análise de Sentimento	(CHANG, 2018)
Metadata extraction and classification of YouTube videos using sentiment analysis	Áudio - Vídeos	Análise de Sentimento	(RANGASWAMY et al., 2016)
Youtube movie reviews: Sentiment analysis in an audio-visual context	Transcrição	Análise de Sentimento	(WÖLLMER et al., 2013)
Research on Video Automatic Feature Extraction Technology Based on Deep Neural Network	-	Extração de Conhecimento	(QING et al., 2021)
Large Scale Open Source Video Recommender Tool Using Metadata Surrogates	Metadado	Sistema de recomendação	(MATHEW; SMITH; PASSARELLI, 2018)

Tabela 3.1: Trabalhos encontrados no mapeamento sistemático e não citados nesta seção.

Capítulo 4

Procedimentos Metodológicos

Neste capítulo são apresentados os procedimentos metodológicos que norteiam esta pesquisa. Conforme apresentado na figura 4.1, esta pesquisa foi dividida em quatro etapas, começando no planejamento inicial, onde foram elaboradas as questões de pesquisa, estruturação da coleta dos dados, desenvolvimento do método até finalmente a obtenção e avaliação dos resultados.

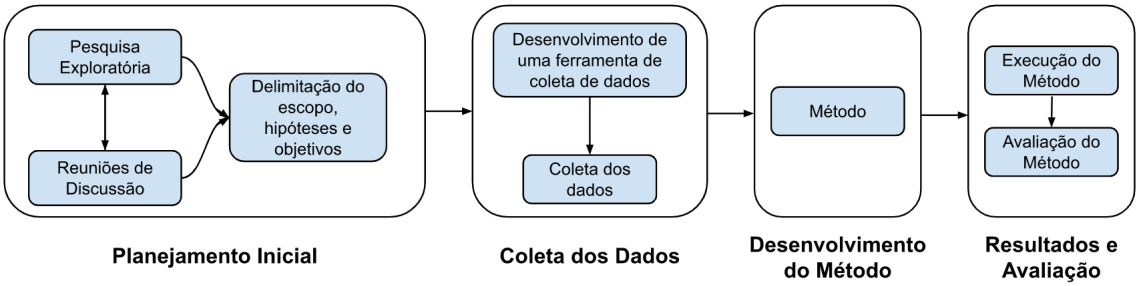


Figura 4.1: Estrutura da Pesquisa

4.1 Planejamento Inicial

Visando o desenvolvimento de um método para a extração de conhecimentos a partir de transcrições de áudio em vídeos, de domínio aberto, elaborou-se uma estratégia para a condução deste trabalho. Nesta etapa, através de reuniões de discussões, algumas "perguntas chaves"foram elaboradas para definirmos os próximos passos. Entre elas, podemos citar:

- Podemos considerar redes sociais baseadas em vídeos uma fonte de conhecimentos confiável (no sentido da qualidade da informação apresentada)?

- Existe alguma característica que difere transcrições de áudio de outras fontes textuais?
- Quais informações são relevantes em um modelo de extração de informação a partir de transcrições de vídeos?
- Como extrair conhecimento de uma transcrição de áudio?
- Qual seria a melhor estrutura de dados para representar o conhecimento?

Com o objetivo de responder esses questionamentos, foi realizada uma pesquisa exploratória e um mapeamento sistemático (mais detalhes no capítulo 3) para obter um melhor entendimento sobre o tema. Além disso, algumas decisões importantes foram tomadas. Primeiramente, foi necessário decidir qual a rede social seria utilizada. O Youtube foi escolhido, por ser a maior rede de compartilhamento de vídeos do mundo (YOUTUBE, 2020), e principalmente por oferecer uma *API* que permite a extração das transcrições de áudio.

Diversos trabalhos provam que as redes sociais baseadas em vídeos podem ser utilizadas como uma ferramenta de aprendizado e geração de conteúdos de qualidade. Alguns autores vão além e utilizam o Youtube como ferramenta de ensino integrada com a sala de aula como em (ZAHN et al., 2014), (ORÚS et al., 2016) e (DIAS, 2013).

Conforme já citado na introdução deste trabalho, os arquivos textuais gerados a partir de transcrições de vídeos, possuem algumas características únicas em relação a outras fontes de dados, como por exemplo a presença de ruídos, erros semânticos e ausência de pontuações nos textos. Além disso, a linguagem utilizada também difere de outras fontes textuais, tendendo a ser mais coloquial. Este fato por si só, pode dificultar o emprego de modelos treinados a partir de textos mais formais (como por exemplo de notícias jornalísticas). Na seção 4.2 é apresentado em maiores detalhes os passos para tratar estas particularidades.

Objetivando entender quais dados seriam ou não relevantes (além das próprias transcrições) para o método proposto, os autores em (MUNARO et al., 2020) demonstram que diversos fatores interferem significantemente na popularidade, engajamento e sucesso de um vídeo. O entendimento destes fatores, constituem um importante conjunto de variáveis que devem ser considerados na construção do grafo de conhecimento proposto. Desta forma é ilustrado, que os seguintes metadados devem ser preservados no modelo: **número de likes/dislikes**, **número de comentários**, **valência** e **duração do vídeo**. Para mais informações sobre este estudo, consulte a seção 6.4.2.

Outro ponto chave a ser decidido, foi em relação a qual técnica de extração de informação seria utilizada. Devido ao fato do método proposto ser de domínio aberto, não seria viável utilizar uma abordagem baseada em regras ou mesmo supervisionada. Decidiu-se então, utilizar um extrator de informações aberto (OpenIE) (vide seção 2.1.2).

Por fim, em relação à estrutura de representação do conhecimento, como já descrito anteriormente, optou-se por trabalhar com grafos de conhecimento, principalmente pela flexibilidade de dispensar a necessidade de um esquema pré-definido, além de permitir, através do protocolo RDF a expansão da base de conhecimento gerada com outros grafos, como o DBPedia, por exemplo.

4.2 Coleta dos dados

Devido à ausência de um corpus na literatura que atenda as demandas desta pesquisa, foi necessário implementar uma ferramenta que extraia as transcrições, bem como, os metadados presentes nos vídeos. Toda a implementação desta ferramenta foi desenvolvida na linguagem de programação Python e a persistência dos dados em um banco de dados relacional MySql. O processo de extração dos dados (com exceção das transcrições) foi feito integrando com API oficial do Youtube (chamada de Youtube Data V3 [1]).

Os dados extraídos são classificados em duas categorias: **informações do canal** e **informações do vídeo**.

Podemos entender um canal como a "página"ou autor responsável por realizar a postagem de um determinado vídeo. Nas informações do canal os seguintes dados são coletados:

- Url do canal: Link que direciona para a página do canal;
- Id do canal: Identificador único do canal fornecido pelo Youtube;
- Categoria: Este campo é preenchido pelo proprietário do canal. Existem diversas categorias disponíveis, como por exemplo, moda, saúde, gaming, etc... ;
- Gênero: gênero do proprietário do canal. Pode ser classificado em masculino, feminino, ambos (quando o canal possui mais de um proprietário) e não se aplica (em canais de desenho animado por exemplo);
- Nome do canal;
- País de origem;

[1] <https://developers.google.com/youtube/v3>

- Ano de criação;
- Descrição do canal: texto produzido pelo proprietário do canal (Normalmente é um resumo do canal);
- Total de vídeos postados;
- Total de inscritos;
- Total de visualizações: somatório de todas as visualizações de todos os vídeos;

Já nos vídeos, as seguintes informações são extraídas:

- Id do video: identificador único do vídeo fornecido pelo Youtube;
- Id do canal;
- Título do vídeo;
- Descrição do vídeo;
- Transcrição do áudio;
- Total de comentários;
- Total de dislikes;
- Total de likes;
- Total de visualizações;
- Duração do vídeo;
- Data da publicação;

Por fim, a ferramenta possui três modos de operação diferentes:

- **Extração por canal:** Neste modo é possível inserir a url de um canal e o sistema irá extrair os dados de todos os vídeos deste canal;
- **Extração por palavra-chave:** Na extração por palavra-chave o usuário insere um conjunto de palavras (comumente chamado de query de busca) e o sistema irá extrair todos os vídeos retornados desta busca;
- **Extração por vídeo:** modo de operação mais simples. Neste modo o usuário insere a url de um vídeo e em seguida é feita a extração;

4.3 Desenvolvimento do Método

A implementação do método permite a realização da conversão de dados brutos em conhecimento, utilizando grafos. O resultado final esperado é a obtenção de um grafo de conhecimento, a partir de diversos subgrafos com diferentes níveis de abstrações. Na figura 4.2 é possível visualizar as camadas de abstração desenvolvidas. É importante ressaltar que cada camada apresentada representa um subgrafo do GC.

A primeira abstração do método é a **camada dos metadados**. Ela é responsável por representar os dados oriundos do Youtube. Desta forma é possível visualizar os dados, tanto dos vídeos como dos canais (números de visualizações, likes/dislikes, total de comentários etc.).

Em seguida, entramos na **camada dos tópicos latentes**. O objetivo dela é classificar os tópicos (domínios) dos vídeos presentes na base. Com isso, podemos relacionar vídeos de diferentes canais, que possuem assuntos correlatos.

Já na **camada das entidades nomeadas**, é extraído todas as entidades nomeadas presentes nas transcrições de áudio. Desta maneira, o método é capaz de relacionar vídeos e canais que abordam de um local, pessoa ou evento em específico.

A última camada do método permite a extração das relações semânticas das transcrições dos vídeos (chamadas de **tuplas de conhecimento**). É nesta abstração, onde "armazenamos os conhecimentos dos vídeos"[2].

O somatório das camadas contribui para a obtenção do grafo de conhecimento.

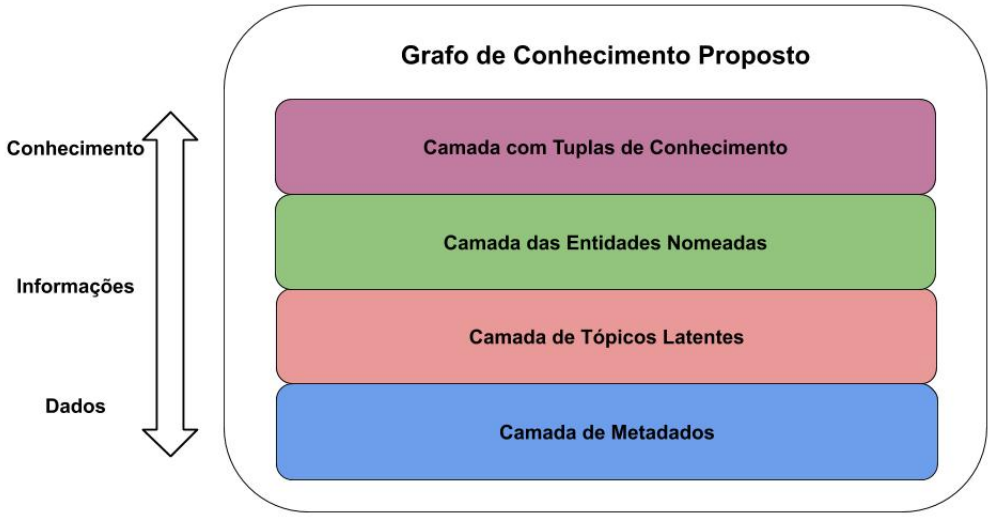


Figura 4.2: Subgrafos do GC

Dado que toda esta pesquisa se baseia na utilização de grafos de conhecimento,

[2]Cabe uma ressalva ao termo conhecimento utilizado nesta camada. O sentido de conhecimento significa persistir os dados extraídos do extrator de informação de aberto.

uma abordagem de persistência de dados relacional não faria sentido neste trabalho. Desta forma, foi necessário encontrar um banco de dados (BD) NOSQL [3] baseado em grafos para realizar a persistência das tuplas geradas no processo de extração de informação.

O Neo4J [4] é um BD, implementado em Java, com uma versão gratuita que permite armazenar e consultar dados através de grafos. Além disso, ele possui uma linguagem própria de consulta (chamada de Cypher), open source, que permite entre outras coisas a execução de algoritmos de aprendizagem de máquina, cálculo de métricas em grafos (grau, distância, centralidade, pageRank, ...) [5] e operações de agregações.

Para o processo de extração de informação, buscou-se uma ferramenta, considerada estado da arte e preferencialmente de código aberto, para o português brasileiro. Foram encontradas as seguintes ferramentas: (OLIVEIRA; GLAUBER; CLARO, 2017), (SENA; CLARO, 2018) e (COLLOVINI; MACHADO; VIEIRA, 2016). Foi definido o uso do extrator de (SENA; CLARO, 2018), por ser uma ferramenta com o código fonte disponível [6] e por apresentar resultados superiores aos demais (conforme experimentos realizados pelos autores da ferramenta).

Para a extração dos tópicos latentes será utilizado o algoritmo LDA (BLEI; NG; JORDAN, 2003), pois este tende a gerar resultados mais legíveis para os humanos (maior coerência) (O'CALLAGHAN et al., 2015).

Toda a implementação do método e dos experimentos foi feita na linguagem de programação **Python** (versão 3.7). Para a geração dos modelos de topic modeling foi utilizada a biblioteca **Gensim** [7], e para realizar o pré-processamento dos dados textuais, o **Spacy** [8]. Já para gerar o conjunto de treinamento e teste dos modelos, empregramos o **Scikit Learn** [9]. Por fim, para a extração da valência (sentimento) dos vídeos, foi utilizada a biblioteca **TextBlob**[10], que um pacote baseado na biblioteca NLTK. No capítulo 5, é apresentado o método que será executado utilizando as tecnologias descritas. Já no capítulo 6 será discutida as características do corpus utilizado e da base de dados gerada a partir dele.

[3] NOSQL é um termo genérico para definir bancos de dados, que suportam outras formas de representação de dados além do formato tabular (relacional). O termo deriva de *Not Only SQL*, que significa o suporte a outras linguagens de consulta além do SQL padrão.

[4] <https://neo4j.com/>

[5] <https://neo4j.com/docs/graph-data-science/current/algorithms/>

[6] <https://github.com/FORMAS/PragmaticOIE>

[7] <https://radimrehurek.com/gensim/>

[8] <https://spacy.io/>

[9] <https://scikit-learn.org/stable/>

[10] <https://textblob.readthedocs.io/en/dev/>

4.4 Resultados e Avaliação

Para que seja possível avaliar os resultados obtidos, bem como determinar a eficácia do método proposto, é necessário definirmos algumas métricas de avaliação. Estas métricas estão separadas em três categorias distintas, que possuem propósitos diferentes:

- **Métricas de Análise de Grafos:** permitem interpretar a topologia dos grafos e seus dados;
- **Métricas de Qualidade de Grafos:** tem como objetivo analisar o "raciocínio"na descoberta de novos padrões e inferências;
- **Métrica de Análise de Tópicos Latentes:** Analisa a qualidade dos tópicos latentes gerados;

4.4.1 Métricas de Análise de Grafos

Conforme apresentado na seção 2.3.2.3, os grafos de conhecimento podem ser analisados a partir da **centralidade**, **comunidades**, **conectividade**, **similaridade** e **análise de caminhos**. Para a análise de centralidade serão utilizados algoritmos degree, closeness e PageRank. Já para a detecção de comunidades será empregado o algoritmo label-propagation. Na conectividade será calculada a densidade dos grafos, e utilizado o algoritmo de Karger's. Para encontrar os nós mais similares, utilizaremos o algoritmo de equivalência estrutural. Estes algoritmos foram escolhidos pois, segundo Hogan e colegas, estas técnicas são comumente utilizadas em avaliações de grafos de conhecimento (HOGAN et al., 2021). No âmbito deste projeto, os algoritmos de centralidade e comunidades podem ser úteis para determinar quais conteúdos e/ou vídeos são relevantes ou virais. Já o cálculo de similaridade pode ser usado para validação de fatos e inferências lógicas.

4.4.2 Métricas de Qualidade de Grafos

Neste trabalho, as métricas de qualidade de grafos visam oferecer o ferramental necessário para avaliar o refinamento (mineração de regras) das tuplas de conhecimento, extraídas a partir de um extrator de informações.

Segundo (SUCHANEK et al., 2019) ao se refinar um GC, dois indicadores são importantes:

- **Suporte:** Número de vezes que uma determinada regra é detectada no GC;
- **Confiança:** Razão que uma determinada regra se prova verdadeira;

Como todo o processo de extração das tuplas de conhecimento é feito de forma não supervisionada determinar se uma regra (tupla) é verdadeira, não é trivial. Assim sendo, é importante também, definir algumas métricas para determinar a qualidade das tuplas obtidas a partir das sentenças de entrada. A pesquisa de (ABREU; BONAMIGO; VIEIRA, 2013) define estas métricas, como sendo a *precisão* (equação 4.1), *medida F1* (equação 4.3), *revocação* (equação 4.2) e *minimalidade* (equação 4.4).

$$\text{Precisão} = \frac{\text{Nº de fatos Válidos}}{\text{Nº de fatos Extraídos}} \tag{4.1}$$

$$\text{Revocação} = \frac{\text{Nº de coerentes extraídos}}{\text{Nº de fatos coerentes}} \tag{4.2}$$

$$F1 = \frac{2 * \text{Precisão} * \text{Recall}}{\text{Precisão} + \text{Recall}} \tag{4.3}$$

$$\text{Minimalidade} = \frac{\text{Nº de fatos Mínimos}}{\text{Nº de fatos Válidos}} \tag{4.4}$$

Neste trabalho, o objetivo do cálculo do suporte e da confiança é permitir uma análise em relação a qualidade das tuplas de conhecimento que estão sendo extraídas. Conforme já explicado anteriormente, os extratores de informações abertos tendem a gerar um grande volume de fatos incorretos, similares e/ou incorretos. Desta forma, o método proposto apresenta uma técnica que visa reduzir estes erros e inconsistências (refinamento do grafo). Assim sendo, os resultados obtidos após o refinamento, serão comparados seguindo um "padrão ouro"(base rotulada por especialistas humanos), calculando-se os suportes e confianças.

Por fim para encontrar as tupla mais representativas a partir dos seus grupos de origem, serão utilizadas as métricas tradicionais de aprendizagem de máquina: acurácia, revocação e medida F1.

4.4.3 Métrica de Análise de Tópicos Latentes

Em (BLEI; NG; JORDAN, 2003) são apresentadas duas métricas para se realizar a análise de tópicos latentes: **perplexidade** e **coerência** (mais detalhes na seção 2.1.3).

Como o objetivo final do grafo gerado é oferecer um conjunto de dados interpretável aos humanos, segundo (HAGEN, 2018), a coerência tende a oferecer melhores resultados. Chang, vai além e afirma que modelos que possuem uma baixa perplexidade (que significa uma boa generalização do modelo), possuem uma correlação negativa com legibilidade

dos tópicos gerados pelos humanos (CHANG et al., 2009). Desta forma, neste trabalho utilizaremos somente a coerência para analisar os tópicos latentes.

4.4.4 Desenvolvimento de uma Base de Avaliação e Testes

Conforme apresentado anteriormente, para avaliar o método desenvolvido, é necessário utilizar uma base de dados com informações corretas e verdadeiras. Pensando nisto foi elaborada uma metodologia a fim de se obter estes dados. Esta metodologia é apresentada na figura 4.3. Este processo tem como objetivo, a partir da extração de tuplas de conhecimento (maiores detalhes na seção 5.7) encontrar: (i) as tuplas (ou "fatos") mais similares entre si e (ii) mapear qual tupla deste conjunto melhor representa este grupo. Para isto, três avaliadores humanos especialistas na área de PLN e *marketing* realizaram a rotulação manual destas tuplas a partir de uma ferramenta própria para este fim. Após isto, foi calculado o coeficiente de correlação intraclasse (em inglês ICC) para determinar o nível de concordância entre os avaliadores (BARTKO, 1966). Caso o coeficiente de concordância fique abaixo de um limiar predefinido, deve-se então realizar uma reunião de alinhamento entre os avaliadores com o intuito de se entender as divergências encontradas. Após isso, novamente todo processo deve ser refeito até ser obtido um resultado acima do limiar estipulado.

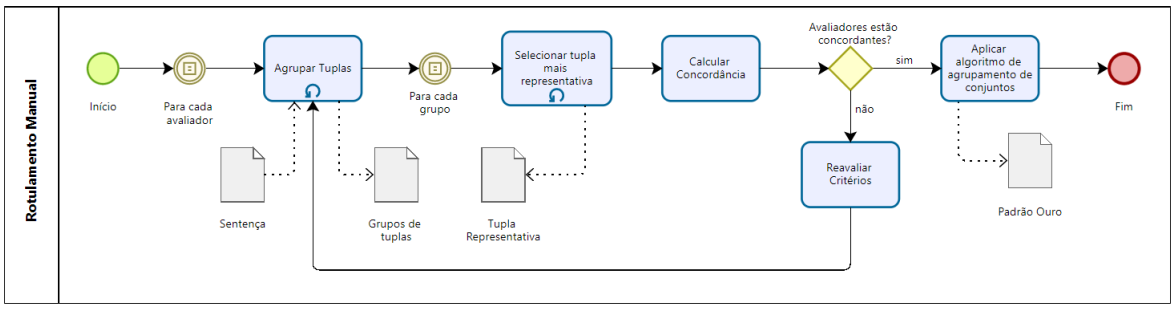


Figura 4.3: Metodologia para obtenção do padrão ouro.

Por fim, finalizada a rotulação pelos especialistas humanos, existirão três conjuntos de dados diferentes a serem processados, sendo um conjunto por especialista. Desta forma é importante aplicar uma técnica de agrupamento de agrupamentos (do inglês clustering clusters ou ainda cluster ensemble) para obter somente um único agrupamento que representa o padrão ouro a ser utilizado nas análises. Isto será feito a partir do algoritmo MCLA (Meta-CLustering Algorithm). O MCLA, foi escolhido por duas razões: (i) por ser um algoritmo baseado em grafos e (ii) por ter apresentado os melhores resultados em testes empíricos (ver seção 6).

Capítulo 5

Método Proposto

Este capítulo apresenta o método proposto nesta pesquisa. A Figura 5.1 oferece uma visão geral do método que será melhor descrito nas seções a seguir.

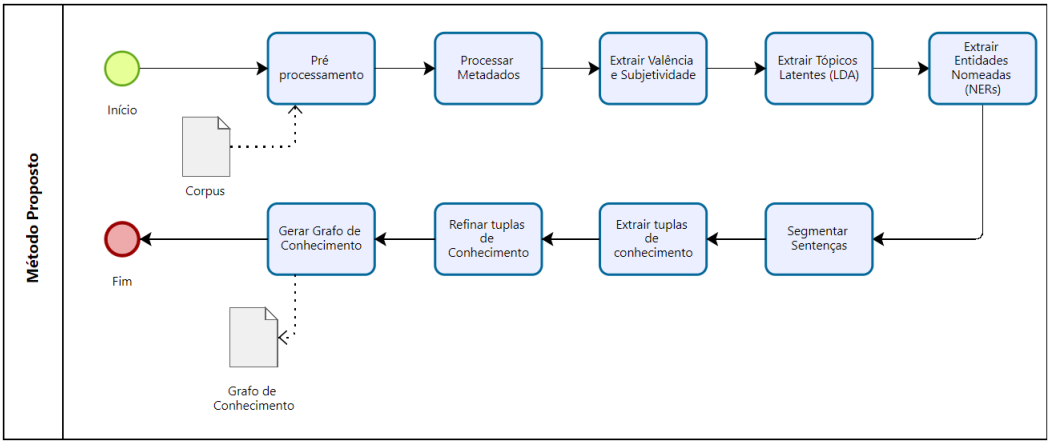


Figura 5.1: Método Proposto

5.1 Pré-processamento

O pré-processamento realiza uma série de operações básicas sobre o dado textual bruto (transcrições de áudio) para permitir uma representação vetorial do mesmo. A figura 5.2 ilustra todos as operações realizadas no pré-processamento.

A primeira atividade a ser realizada é a remoção das *stop words* e transformar todos os caracteres maiúsculos em minúsculos. Neste processo, foi utilizado a lista padrão fornecida pelo Spacy [1] expandida com as stop words apresentadas no anexo 8.1. As palavras adicionadas no anexo 8.1 podem ser resumidas em três categorias principais:

[1]Stop Words: <https://github.com/explosion/spaCy/blob/master/spacy/lang/pt/stop_words.py>

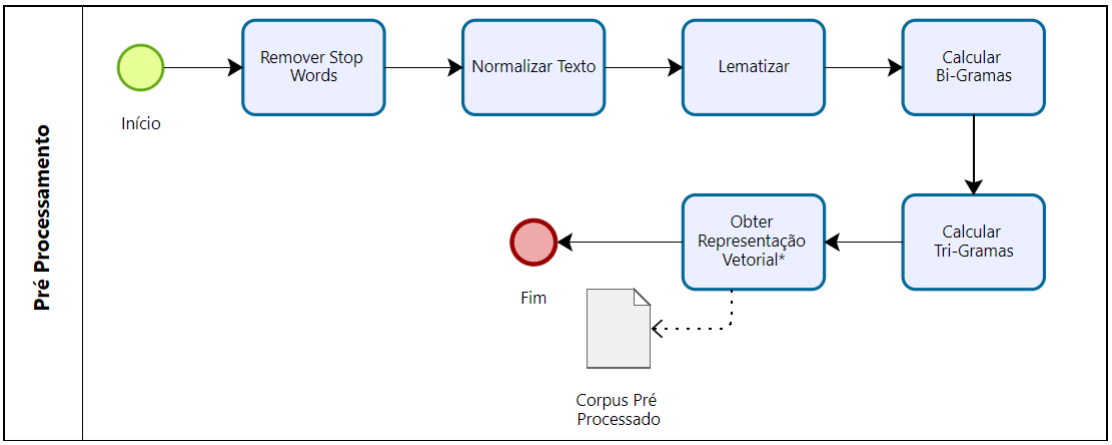


Figura 5.2: Processo do pré-processamento

- **Tags de Marcação**: Tags adicionadas automaticamente pelo Youtube no processo de speech recognition. Elas são responsáveis por marcaram momentos em que alguma música toca no vídeo, ou momentos de aplausos. Elas são sempre delimitadas por um "["de abertura e um "]"de fechamento. Exemplos: "[APLAUSOS]", "[MÚSICA]";

- **Palavras referentes ao contexto do Youtube**: Palavras que remetem a ações e comportamentos dentro da rede. Exemplo: "Compartilhar", "Seguir", "Like", "Joinha";

- **Palavras frequentemente utilizadas na linguagem falada coloquial**: Vícios de linguagem orais (principalmente cacoetes). Exemplos: "Tipo", "Né", "Aí";

Após a remoção das stop words, foi aplicado um lematizador, com o objetivo de diminuir a dimensionalidade dos dados, reduzindo todos os termos ao seu lemma (palavra na forma infinitiva, masculina e no singular). Adicionalmente a isso, é identificado todos os bi-gramas e tri-gramas presentes no corpus (segundo Blei, Ng e Jordan (2003), esta etapa é importante para melhorar os resultados do modelo LDA de topic modeling). Dois parâmetros são importantes para o cálculo do n-gramas: ***min_ count*** e ***threshold***. O *min_ count* é responsável por determinar quais termos são elegíveis para formar os gramas. Somente serão aceitos termos que possuem uma frequência maior que este parâmetro. Após a geração dos bi e tri-gramas o *threshold* determina quais os gramas serão mantidos na base processada. Todo grama gerado que tenha uma frequência menor que o *threshold* é descartado. Após testes empíricos, determinou-se um *threshold* de 50 e um *min_ count* de 5 (RODRIGUES; PARAISO, 2020).

Finalmente, a última tarefa do pré-processamento é a obtenção das representações vetoriais dos documentos, termos (*tokens*) e sentenças. Estas representações serão obtidas

através de cinco abordagens diferentes:

- **Bag of Words**;
- **Bag of Words + TF-IDF**;
- **Word Embeddings:** Word2Vec e BERT (Bidirectional Encoder Representations from Transformers);

Em relação aos *Word Embeddings*, a escolha do BERT se justifica por ser um modelo atual e considerado estado da arte pela literatura em tarefas de PLN (DEVLIN et al., 2018a). Entretanto, apesar do BERT entregar bons resultados ele é um modelo robusto, com alta dimensionalidade (mais de 700 dimensões) e consequentemente possui um elevado custo computacional no seu processamento. Desta forma foi selecionado ainda o Word2Vec, que também é extensivamente utilizado pela literatura. Por fim, o Bag of Words será utilizado para servir como uma referência (baseline) entre os resultados. Além disto, na etapa de modelagem de tópicos (seção 5.4) a única representação vetorial utilizada, será o Bag of Words em função de uma limitação do LDA não trabalhar com *Word Embeddings* (BLEI; NG; JORDAN, 2003).

Conforme já apresentado no capítulo 2, o Word2Vec é um modelo capaz de gerar representações numéricas de palavras utilizando redes neurais, capturando o significado semântico das palavras e as relações entre elas (MIKOLOV et al., 2013a). Neste trabalho, foi utilizado um modelo Word2vec treinado para língua portuguesa a partir da biblioteca Spacy [2], que fornece vetores densos de 300 dimensões.

Para geração das representações contextuais, foi utilizado o modelo de linguagem BERT (DEVLIN et al., 2018b) versão multilíngue, que oferece suporte para mais de 100 idiomas incluindo português. O BERT utiliza o codificador da arquitetura Transformer (VASWANI et al., 2017), que por sua vez faz uso do mecanismo de atenção ao aprender as relações contextuais entre palavras em um texto. Com BERT, é possível gerar representações de palavras contextuais e bidirecionais, ou seja, o codificador lê toda a sequência de palavras de uma vez e não apenas da direita para esquerda, de forma a aprender o contexto de uma palavra com base em toda sua vizinhança. Neste trabalho, geramos as representações de cada tupla utilizando vetores densos de 768 dimensões, gerados com a biblioteca NLU da *John Snow Labs*[3].

[2] <https://spacy.io/usage/linguistic-features#vectors-similarity>
[3] <https://www.johnsnowlabs.com/>

5.2 Processamento dos Metadados

Nesta seção é definido o esquema de um sub-grafo para representar os vídeos e canais presentes na base de dados. O formato deste sub-grafo será do tipo *RDF*, podendo cada nó representar um canal ou vídeo. O identificador único (URI) de cada nó é o próprio identificador do vídeo ou canal.

Desta forma, esta etapa do método, é responsável por organizar e agrupar os dados colhidos pela API oficial do Youtube, juntamente com as transcrições de áudio na forma de um grafo. Todos dados coletados serão persistidos no banco de dados não relacional *Neo4J*. Na Figura 5.3, temos um esquemático da composição deste sub-grafo. Analisando a figura de exemplo, podemos notar os primeiros benefícios de uso de estrutura de dados baseadas em redes. Entre eles podemos citar:

- **Abstração de conceitos:** no lugar de dados brutos, podemos trabalhar com os conceitos que estes dados representam, representando-os a partir de entidades (Canal e Vídeo);
- **Facilidade de implementação:** Caso fosse utilizado alguma representação relacional destes dados, seria necessário a modelagem de diversas tabelas para este fim;
- **Agilidade na recuperação da informação:** devido a possibilidade de realizar pesquisas a partir de conceitos, o processo pela busca de informações é facilitado. Ademais, é possível também, buscar relacionamentos entre diferentes conceitos e entidades. Por exemplo, é possível realizar consultas para encontrar os canais mais relevantes em uma dada categoria ou ainda encontrar os canais que mais realizam parcerias entre si (a partir da publicação de vídeos em conjunto).

Por fim, é importante mencionar que este grafo será unificado com as demais informações que serão coletadas e/ou transformadas nas seções a seguir, para finalmente obtermos um grafo de conhecimento completo (ver seção 5.9).

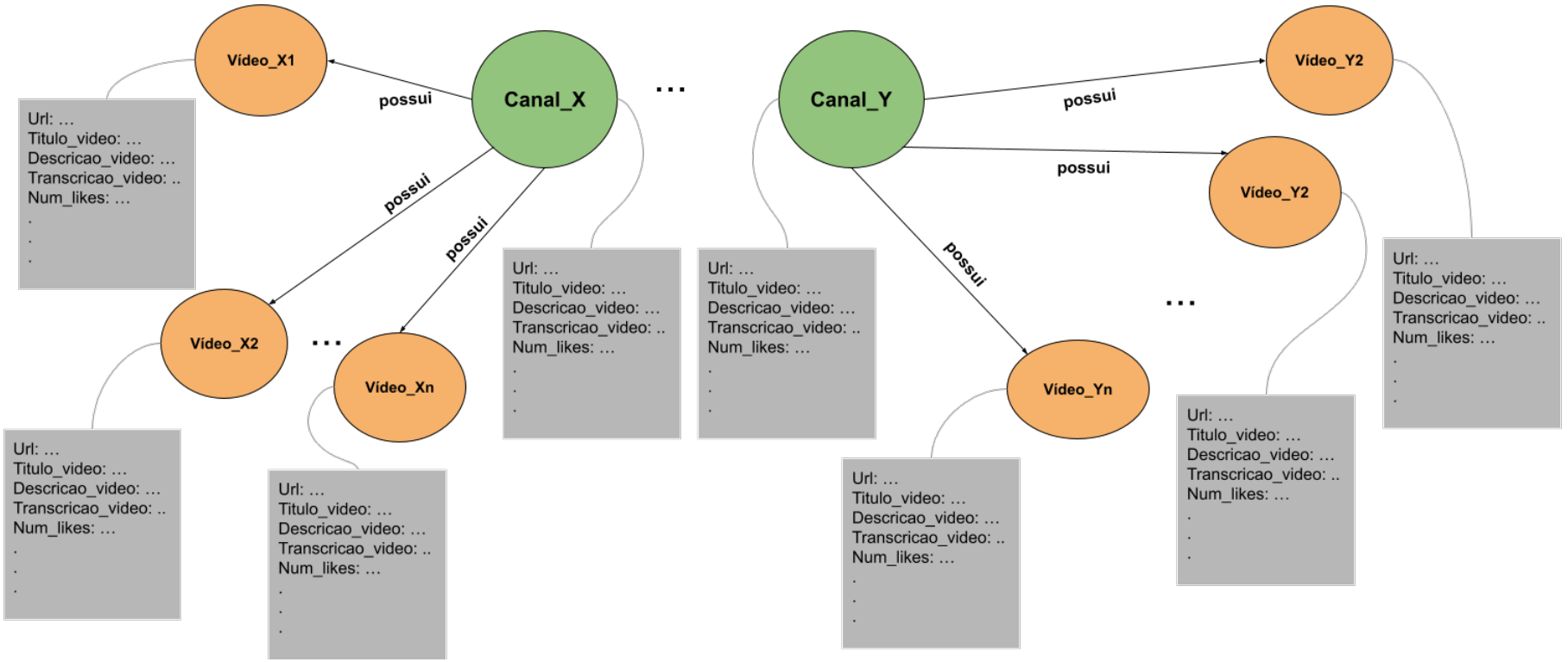


Figura 5.3: Demostração do Sub-Grafo dos Metadados

5.3 Extração da Valência e Subjetividade

Esta etapa do método é responsável por extrair o sentimento presente nos vídeos que compõem a base dados. Utilizando a biblioteca *TextBlob*, serão calculados a polaridade e a subjetividade dos autores dos vídeos.

Entende-se como polaridade, o teor emocional de um documento, possuindo como intervalos o valor de -1 como um documento extremamente negativo, zero neutro e +1 como um teor altamente positivo.

Vale mencionar que o modelo utilizado pelo *TextBlob* para a extração de sentimentos é baseado no léxico de palavras *SentiWordNet*, e para classificação dos documentos foi utilizado o algoritmo NaiveBayes (SEBASTIANI; ESULI, 2006)[4].

5.4 Extração dos Tópicos Latentes

Neste trabalho foi utilizado o algoritmo de extração de tópicos latentes LDA. Este algoritmo possui como limitação a impossibilidade de inferir o número ideal de tópicos de um corpus (chamado de k). Desta forma, este número deve ser definido antes da sua execução. Uma abordagem comum para inferir o número ideal de tópicos, é gerar diversos modelos com diferentes valores de k e com o auxílio de um especialista analisar os resultados obtidos, até encontrar um resultado satisfatório (SOUZA; SOUZA, 2019). Desta forma, foi necessário desenvolver uma abordagem que permitisse encontrar o melhor número de k em um corpus de forma automatizada, conforme demostrado na figura 5.4.

O algoritmo 1 apresenta o processo que está sendo proposto para a obtenção automática do número "ideal"de tópicos. O método gera um conjunto de n modelos de forma a ir incrementando o número de k tópicos. Após a criação de cada modelo, é também calculada coerência deste modelo a partir da média da coerência dos tópicos. Este cálculo será importante por duas razões: (i) determinar o critério de parada na geração de novos modelos e (ii) auxiliar na seleção do melhor modelo. Para determinar o critério de parada da geração dos modelos incrementais, é calculado o coeficiente de correlação de Pearson entre a coerência dos modelos e o seu número de tópicos. Este valor será responsável por determinar se as coerências dos modelos seguem uma tendência alta ou queda em relação ao incremento dos k tópicos. Desta forma o cálculo da correlação será feito a partir de janelas deslizantes (jd) pré-definidas. A partir do momento que for detectada uma correlação negativa (tendência de queda) nos modelos que compõe a jd, o treinamento se

[4]Código fonte responsável pelo modelo de análise de sentimentos: <https://github.com/sloria/TextBlob/blob/90cc87ab0f9e25f37379079840ec43aba59af440/textblob/en/sentiments.py>

encerra.

A próxima etapa a ser executada é, selecionar um candidato para ser considerado o melhor modelo a partir da lista dos modelos obtidos e que consequentemente, possua o número mais adequado de tópicos que represente o corpus utilizado. Isso é feito, selecionando o modelo com a maior coerência disponível. Por fim, a última etapa a ser executada, é realizar um cálculo da diferença entre o modelo candidato e os modelos que possuam um menor número de k e uma coerência similar. Entende-se como uma coerência similar valores com uma diferença inferior a 2% da maior coerência encontrada. Este processo é executado pois, após testes experimentais (mais detalhes na seção 6.2), observou-se que modelos com um maior k e uma coerência similar possuíam uma menor representatividade na classificação dos documentos, de forma que, diversos tópicos não representavam um "domínio de assuntos" significativo, sendo desta forma, composto majoritariamente de termos genéricos.

Algoritmo 1: Método proposto para a seleção dinâmica do melhor modelo LDA

```
    input  : janela_deslizante: Inteiro que determinará o tamanho da janela a ser utilizada para
             o cálculo da correlação
[1] Function obter_melhor_modelo(janela_deslizante):
[2]     while correlacao > 0 do
[3]         num_topicos +=1 ;
[4]         modelo = treinar_modelo_lda(num_topicos)
[5]         if (num_topicos % janela_deslizante)==0 then
[6]             correlacao = calcular_correlacao(lista_modelos[janela_deslizante:]);
[7]     melhor_modelo = selecao_modelo(lista_modelos)
[8]     return melhor_modelo

    input  : lista_modelos: lista de objetos contendo os modelos, coerência e número de tópicos
[9] Function selecao_modelo(lista_modelos):
[10]    maior_corencia = max(lista_modelos.coerencia);
[11]    lista_corencia_similar = list(filter(lambda x: (maior_corencia - x)<=0.02,
          lista_modelos.coerencia))
[12]    melhor_modelo = min(lista_corencia_similar.num_topicos)
[13]    return melhor_modelo
```

Por fim, após a obtenção do modelo mais apropriado, todos os documentos do corpus são classificados com este modelo a fim de que cada vídeo da base esteja incluído em um ou mais tópicos. Em outras palavras, cada transcrição será inserida no modelo a fim do modelo inferir quais tópicos melhor descrevem cada documento.

5.5 Extração das Entidades Nomeadas

A extração das entidades nomeadas (NERs), é uma importante tarefa na construção do GC, por permitir a representação de conceitos como, pessoas, marcas, empresas e

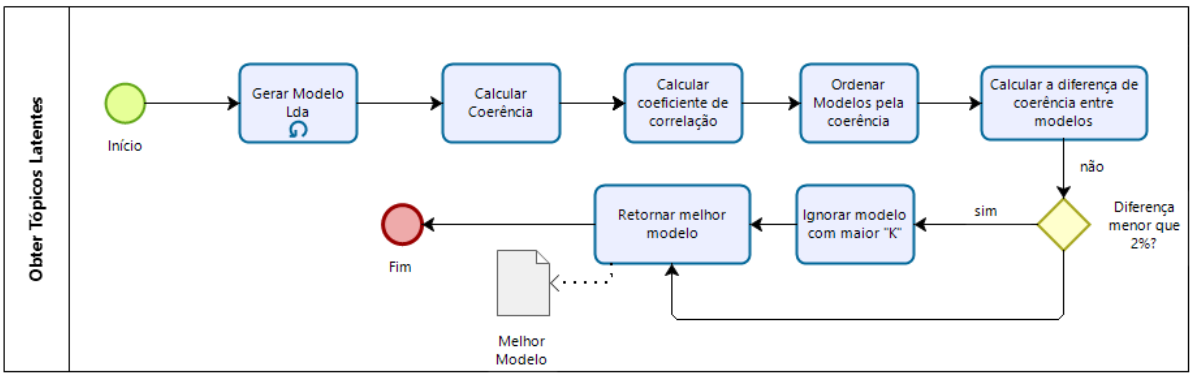


Figura 5.4: Extração dos tópicos latentes

localizações presentes nos vídeos.

Desta forma, integrando as NERs obtidas com o GC que será gerado (seção 5.9), será possível consultar vídeos que citam uma determinada pessoa e/ou marca por exemplo.

Para a extração das entidades foi utilizada a biblioteca *Spacy*, utilizando o modelo pré treinado fornecido pela biblioteca, composto por um corpora formado por notícias jornalísticas e dados da Wikipedia [5]. No anexo 8.1, temos a lista completa de todos os rótulos das entidades nomeadas possíveis.

Cada entidade nomeada identificada em um vídeo será representada na forma de uma tupla, vinculando o tipo da NER o conteúdo da NER e o video de origem, conforme apresentado a seguir:

$$tupla\_ner = (documento\_id,\ tipo\_entidade,\ entidade)$$

onde:

$$documento\_id = id\ do\ vídeo\ processado$$
$$tipo\_entidade = rótulo\ da\ entidade$$
$$entidade = nome\ da\ entidade$$

Na Tabela 5.1, temos um exemplo da composição da geração de algumas tuplas a partir de uma transcrição.

5.6 Segmentação de sentenças

Este módulo, é responsável por segmentar as transcrições dos vídeos em suas respectivas sentenças. Isto é importante pois, os extratores de informação devem receber como parâmetro de entrada uma sentença. Caso um documento inteiro fosse passado

[5] <https://spacy.io/models/pt>

Título do Vídeo: MOTOROLA EDGE e EDGE PLUS lançados, Galaxy Note 20 com câmera na S Pen e + | Plantão TC #36		
Url: https://www.youtube.com/watch?v=-GEepA7uSDg&ab_channel=TudoCelular		
Trecho de transcrição: ...P40 no Brasil, Xiaomi trazendo o Redmi Note 9S em menos de um mês após seu lançamento global, visual do Galaxy Note 20...		
Id documento (video id)	**Tipo Entidade**	**Entidade**
-8x4_xJC_Vw	LOC	Brasil
-8x4_xJC_Vw	ORG	Xiomi
-8x4_xJC_Vw	MISC	Redmi Note 9S
-8x4_xJC_Vw	ORG	Galaxy Note

Tabela 5.1: Exemplos de NERs extraídas e sua representação na forma de tupla

para o algoritmo, além do elevado custo computacional em sua execução, o extrator poderia se confundir misturando fatos de diferentes sentenças.

Em PLN, a segmentação de sentenças é uma tarefa de classificação. Tradicionalmente este problema era resolvido a partir de sistemas baseados em heurísticas (*rule-based*). Mais recentemente, alguns autores veem trabalhando com redes neurais nesta tarefa (MARAJ; MARTIN; MAKREHCHI, 2021). Entretanto, independente da abordagem utilizada, os segmentadores de sentenças ainda necessitam da presença de pontuações para realizar o seu trabalho. Conforme já explicado anteriormente, as transcrições ASR carecem de pontuações. Consequentemente, não seria possível gerar sentença alguma nos documentos. Pensando nisto, o método apresentado neste trabalho propõe duas abordagens diferentes para obter as sentenças dos textos: uma para transcrições manuais e outra para transcrições ASR.

Para o processamento das transcrições manuais será utilizada a biblioteca *Spacy*, que irá classificar as sentenças utilizando um sistema de heurísticas[6].

Já para o processamento das transcrições ASR, introduziremos o conceito de *pseudo-sentença* (RODRIGUES; MUNARO; PARAISO, 2021). A pseudo-sentença tem como objetivo permitir a obtenção de um conjunto de palavras que possam ser utilizados no extrator de informação aberto. O primeiro passo para a obtenção de uma pseudo-sentença é a identificação de todas as entidades nomeadas presentes nos vídeos. Isto é importante pois, partiremos do pressuposto que as entidades nomeadas possuem um elevado teor semântico em documentos textuais. Assim sendo, o objetivo das pseudo-sentenças é tentar capturar estes significados. A Figura 5.5, demonstra o fluxo para a geração destas sentenças.

Conforme podemos notar, após a identificação das entidades nomeadas e segmentação de todas as sentenças manuais, foi calculada a média de tokens presentes nestas

[6] <https://spacy.io/api/sentencizer>

sentenças. A média de tokens obtidas nas sentenças manuais serve como uma solução alternativa para montar uma sentença em torno de uma NER. Desta forma, serão calculadas duas médias distintas: a médias de termos a esquerda da entidade nomeada e a média a direita. A Figura 5.6 mostra um exemplo hipotético de uma média de 5 tokens a direita e esquerda e como ficaria a composição de uma pseudo-sentença.

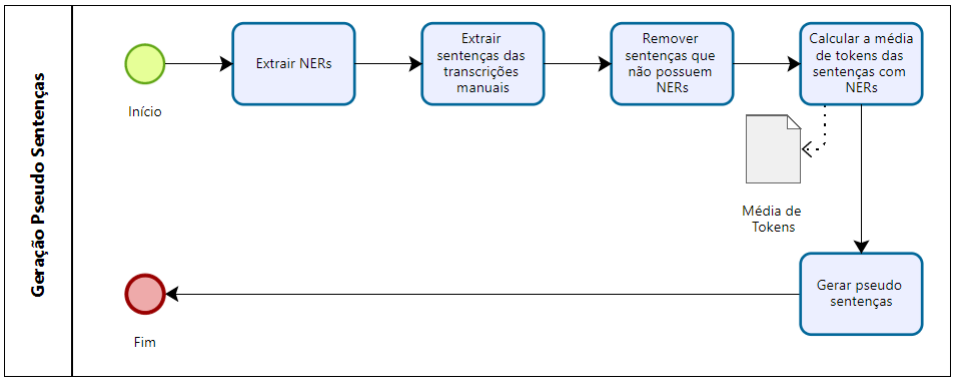


Figura 5.5: Geração das pseudo sentenças.

Figura 5.6: Geração das pseudo sentenças.

5.7 Extração das Informações Abertas (Tuplas de Conhecimento)

Para extrair as tuplas de conhecimento da base de dados, foi utilizado o algoritmo proposto em (SENA; CLARO, 2018). Dado um documento de entrada, este algoritmo é responsável por extrair as tuplas de conhecimento destas sentenças. As tuplas extraídas possuem o seguinte formato: $tupla = (tripla, video\_id, sentenca\_id, tupla\_id)$ onde, $tripla = (arg1, rel, arg2)$ sendo rel a relação semântica que vincula os argumentos 1 e 2 que foram extraídos.

Todas as informações extraídas nesta etapa serão armazenadas em um banco de dados relacional a fim de facilitar o processo do refinamento das tuplas (mineração de regras), que será apresentado a seguir e futuramente utilizado na geração do grafo de conhecimento.

5.8 Refinamento das Tuplas de Conhecimento

Os extratores de informação abertos, oferecem diversas vantagens ao compararmos aos extratores fechados, que são particularmente interessantes ao escopo deste projeto. Entre as principais vantagens podemos citar a flexibilidade em extrair conhecimentos em documentos de domínio aberto.

Entretanto, esta categoria de extratores ainda possui desvantagens, como por exemplo o alto número de fatos extraídos incorretamente, o grande volume gerado de fatos (que inviabiliza a sua interpretação), além de fatos altamente similares e pouco representativos (VO; BAGHERI, 2019). A Tabela 5.2 demostra um exemplo onde ocorrem tais comportamentos. Segundo a tabela apresentada, podemos notar que as tuplas de id 1 ao 4 compartilham similaridades entre si ao passo que a tupla de id 5 não. Desta forma, poderíamos agrupar estas tuplas em dois grupos distintos. Indo além, analisando o grupo 1 com mais profundidade, podemos notar que, apesar das tuplas geradas serem similares, somente a tupla de id 2 foi extraída de forma satisfatória.

Texto entrada: "Eu preciso de um bom nível de ômega 3 como eu disse, que daí já é alguma gordura."				
Id	**Arg1**	**Rel**	**Arg2**	**Grupo**
1	Eu	preciso	de um bom nível de ômega	1
2	Eu	preciso	de um bom nível de ômega 3	1
3	Eu	preciso	de um bom nível 3	1
4	Eu	preciso	de um bom nível de ômega 3 como	1
5	Eu	disse	como	2

Tabela 5.2: Exemplo de tuplas de conhecimento extraídas a partir de um extrator de informações aberto.

Desta forma, esta seção do método se compromete a resolver três desafios:

- **Agrupar tuplas similares;**
- **Encontrar a tupla que melhor representa um dado grupo;**
- **Avaliar se a tupla apresentada é semanticamente correta.**

5.8.1 Agrupamento de Tuplas Similares

Para obter as tuplas mais semânticas a partir de um conjunto de fatos, é necessário encontrar quais tuplas são mais similares entre si a partir de uma dada sentença. Assim sendo, propomos um conjunto de técnicas para encontrar quais conjuntos de tuplas são

mais semelhantes, a partir de duas técnicas distintas: utilizando-se de um algoritmo de agrupamento (*K Means*) e outra valendo-se somente de cálculos vetoriais (distância dos cossenos)[7].

O processo de agrupar tuplas similares passa por dois processos distintos: (i) determinar o número de "tuplas candidatas"e (ii) agrupar as tuplas partir do número de "tuplas candidatas".

Podemos entender a métrica das tuplas candidatas como o número de grupos (k) que melhor representa um conjunto de instâncias de tuplas. Em outras palavras, este valor tem como objetivo informar quantas tuplas distintas melhor representa uma sentença. Seguindo o exemplo apresentado na tabela 6.8, podemos inferir que somente duas tuplas são capazes de representar um conjunto de 5 tuplas.

Na abordagem agrupamento a partir de aprendizado não supervisionado, esta inferência é feita a partir da identificação do pseudo melhor K (número de vizinhos) utilizado no algoritmo de agrupamento (*K-Means*). Como o *K-Means* necessita que o número de grupos (K) seja informado a priori, foi utilizada a técnica da "descida do joelho"(ou "descida do cotovelo"), para determinar quantos grupos melhor representam um dado conjunto de dados ou instâncias (chamado de *pseudo k ótimo*) (SYAKUR et al., 2018). O método da "descida do cotovelo"baseia-se na geração de n modelos de agrupamento. Estes modelos serão treinados com diferentes valores de k (do $k = 2$ até o $k = 100$). E, para cada modelo gerado, calcula-se a pontuação de comparação entre modelos (como por exemplo a métrica da distorção), com isso, será determinado o modelo que melhor descreve os dados (melhor k, ou melhor divisão em grupos).

Este processo é repetido diversas vezes a fim de garantir o melhor $pseudo-k$. Cada modelo foi gerado 10 vezes com diferentes sementes aleatórias de inicialização, gerando 10 possíveis k-ótimos para cada conjunto de dados. Depois, o k mais frequente dentre os 10 candidatos será considerado o $k - pseudo$ ótimo daquele conjunto de dados. O algoritmo 2 demonstra o fluxo para a obtenção do valor do número das tuplas candidatas. Como parâmetros de entrada este algoritmo recebe o atributo *componentes* como uma lista de tuplas em sua representação vetorial, *metric* sendo a métrica utilizada para o cálculo da descida do cotovelo, e max_k sendo o número máximo de grupos que poderão ser criados.

Por fim, uma vez obtidos os pseudos k ótimos, todo o conjunto de tuplas é novamente processado pelo *K Means* com o objetivo de ser obter as composições dos grupos de tuplas a partir de uma sentença de entrada.

[7]Nota ao leitor: Cronologicamente, inicialmente foram realizados experimentos utilizando o algoritmo K-Means. Infelizmente, esta abordagem não gerou resultados satisfatórios. Desta forma, também foi desenvolvida um método a partir de cálculos vetoriais que geraram melhores resultados. A seção 6.3 apresentará estes resultados e discussões sobre o tema.

Algoritmo 2: Agrupamento de tuplas similares a partir de aprendizado não supervisionado;

```
   input  : componentes: lista das representações vetoriais dos documentos;
            metric: algoritmo responsável por computar a descida do cotovelo;
            max_k: número máximo de grupos;
[1]Function descida_cotovelo(janela_deslizante, metric, max_k):
[2]    if len(componentes)==1 then
[3]        return 1
[4]    if len(componentes)==2 then
[5]        return -1
[6]    if len(componentes)<=max_k then
[7]        max_k = len(componentes)
[8]    lista_k = [ ]
[9]    for i = 0, i <= 10, i++ do
[10]       model = KMeans(k=(2,max_k), data=componentes, metric=metric)
[11]       valor_k = model.elbow_value
[12]       lista_k.append(valor_k)
[13]   melhor_k = most_frequent(lista_k)
[14]   if melhor_k is None then
[15]       melhor_k = -1
[16]   return melhor_k
```

Entendido o funcionamento do agrupamento baseado em aprendizado não supervisionado, agora será apresentado uma técnica de agrupamento baseado em cálculo vetorial. O conceito desta técnica se baseia no cálculo da similaridade dos cossenos apresentado na equação 5.1, onde A e B, são duas instâncias a serem comparadas. Caso a *similaridade* entre elas seja maior que um limiar pré-definido elas serão consideradas como pertencentes ao mesmo grupo. Este processo será feito n vezes sendo n o número de tuplas presentes em uma sentença s.

O cálculo da similaridade será aplicado a partir do algoritmo apresentado em 3. Foram considerados como índices de similaridade os valores de 0.7 e 0.8, obtidos empiricamente, no qual se duas tuplas possuem índice de 0.7 ou 0.8 são agrupadas em uma só. No total, foram realizados quatro processamentos: (i) Cálculo de similaridade com representações obtidas do algoritmo Word2vec e limiar de 0.7, (ii) Cálculo de similaridade com representações obtidas do algoritmo Word2vec e limiar de 0.8, (iii) Cálculo de similaridade com representações obtidas do algoritmo BERT e limiar de 0.7, (iv) Cálculo de similaridade com representações obtidas do algoritmo BERT e limiar de 0.8.

Após o processamento, as tuplas com similaridade acima do limite estabelecido foram agrupadas, reduzindo o número de tuplas candidatas inicialmente extraídas.

$$similaridade = \frac{A \bullet B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{5.1}$$

Algoritmo 3: Método proposto para agrupar tuplas similares através da distância de cosseno

```
   input  : sentencas: lista das sentencas extraídas dos documentos;
            threshold: índice mínimo (limiar) para que duas tuplas sejam consideradas similares;
[1] Function agrupa_por_similaridade(sentencas, threshold):
[2]     for sentenca in sentencas do
[3]         for tupla in sentenca.tuplas do
[4]             if len(sentenca.grupo)==0 then
[5]                 grupo = cria_grupo() insere_tupla_grupo(grupo, tupla)
[6]             else
[7]                 for grupo in sentenca.grupos do
[8]                     embedding_tupla = tupla.embeddings
[9]                     for tupla_grupo in grupo.tuplas do
[10]                        embedding_tupla_grupo = tupla_grupo.embeddings
[11]                        indice_similaridade =
                              calcula_similaridade_cosseno(embedding_tupla,
                              embedding_tupla_grupo)
[12]                        if indice_similaridade>=threshold then
[13]                            insere_tupla_grupo(grupo, tupla)
[14]                        else
[15]                            novo_grupo = cria_grupo() insere_tupla_grupo(novo_grupo,
                                  tupla)
```

5.8.2 Seleção da Tupla mais Representativa de um Dado Grupo

Uma vez determinado o número de tuplas candidatas e gerado os agrupamentos destas tuplas, a próxima tarefa a ser executada é determinar qual tupla melhor representa um dado grupo.

Para isto, propomos uma abordagem a partir do cálculo dos centroides destes grupos. De acordo com Seinbach e colegas, um centroide é comumente definido como uma instância média ou mediana das instâncias componentes de um grupo e utilizado como sumário de dados destes grupos na tarefa de agrupamento (STEINBACH; ERTÖZ; KUMAR, 2004). Partindo desta premissa inicial, podemos assumir que as tuplas mais próximas aos centroides dos grupos, melhor carregam a semântica deste conjunto de tuplas agrupadas. Assim sendo, este método propõe como seleção da tupla mais representativa a instância mais próxima ao centroide do grupo. Caso duas distâncias sejam equidistantes ao centro de massa do grupo, a seleção será feita de forma randômica entre os pontos equidistantes do grupo.

5.9 Geração do Grafo de Conhecimento

De posse de todas as tuplas extraídas nas etapas anteriores, pode-se gerar o grafo de conhecimento. Para que seja possível persistir as tuplas no banco de dados e consequentemente gerar o grafo, é necessário converter cada tupla para a linguagem de consulta (*cypher*) do *Neo4j*. Este processo é feito a partir de um *parser* implementado para este fim. A explicação deste *parser* foge do escopo deste trabalho, entretanto vale mencionar que este código é responsável por converter os dados que estão persistidos em um banco relacional *MySQL* para o formato baseado em grafos do *Neo4J*.

Conforme apresentado na seção 4.3 o GC proposto possui quatro camadas de abstração. Para representar estas abstrações serão utilizados cinco tipos de nós diferentes, sendo eles:

- **Nó de Canal**: Representa um canal do Youtube e suas propriedades (total de inscritos, views, data de criação, ...)
- **Nó de Vídeo**: Representa um vídeo postado por um canal;
- **Nó Ie**: Possui os argumentos extraídos com o extrator de informação;
- **Nó Ner**: Entidades nomeadas extraídas dos vídeos;
- **Nó Tm**: Nós que representam os tópicos criados pelo algoritmo de topic modeling;

O objetivo final de todo o método apresentado é obter um grafo de conhecimento similar ao grafo da figura 5.7.

Os nós de *vídeos* (nós em amarelo) e *canais* (nó cinza) são oriundos do processamento dos metadados (seção 5.2). Já os nós do tipo *ners* (roxo) vem das tuplas das entidades nomeadas (seção 5.5). Os nós *tm* (verde) se originam da criação dos tópicos latentes (seção 5.4). E, por fim, os nós *Ie* (marrom) são criados a partir da extração das tuplas de conhecimento (seção 5.7).

Uma vez gerado o grafo de conhecimento, esta estrutura de dados poderá ser utilizada para realizar as mais diversas consultas relacionadas aos temas dos vídeos, como também relacionadas ao produtores de conteúdos (influenciadores digitais).

A lista a seguir demostra algumas possibilidades de consultas que podem ser feitas:

1. Quais os assuntos mais relevantes que estão sendo discutidos e produzidos no Youtube?;
2. Quais autores se posicionam favoravelmente ou negativamente sobre um determinado tema?

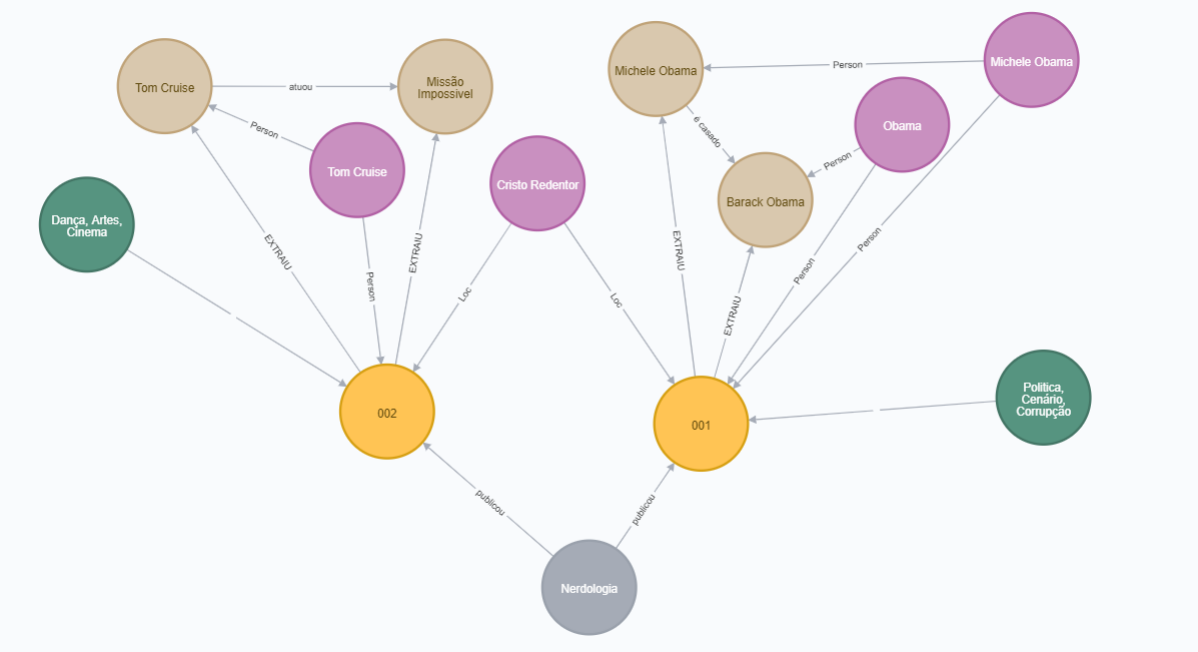


Figura 5.7: Exemplo de um grafo de conhecimento no Neo4j

3. A opinião expressada por um influenciador é condizente com a opinião do público?

4. Visando aumentar o engajamento com o seu público, como um influenciador deve se portar (no sentido de características linguísticas)? Deve agir de modo mais formal ou coloquial por exemplo?

Vale ressaltar, que esta lista mostra somente algumas sugestões de aplicações que podem ser desenvolvidas com o uso do grafo de conhecimento proposto. Desta forma ela não deve ser tratada como uma lista exaustiva das potencialidades do GC.

Finalmente, o capítulo a seguir irá demonstrar os resultados obtidos na construção do grafo de conhecimento descrito, bem como algumas aplicações práticas dele.

Capítulo 6

Resultados

Neste capítulo serão apresentados os resultados obtidos a partir de experimentos realizados. Desta forma, este capítulo será dividido da seguinte forma:

- Obtenção de uma base de dados;
- Um método de modelagem de tópicos em transcrições de vídeos do Youtube;
- Um método para extração e refinamento de tuplas de conhecimento;
- Aplicações práticas de GC em vídeos do Youtube;

6.1 Obtenção de uma base de dados

Neste trabalho foram desenvolvidas três bases de dados distintas: uma com transcrições de vídeos em inglês (*EN-US*), outra com transcrições em português brasileiro (*PT-BR*) e finalmente a última com uma base rotulada manualmente com o agrupamento das tuplas de conhecimento extraídas a partir de um extrator de informação aberto (conforme explicado na seção 4.4.4)[1].

Na versão em inglês da base, foram extraídos 11.177 vídeos de 150 canais diferentes, distribuídos em 11 categorias, conforme apresentado na tabela 6.1. Estes canais foram selecionados por serem considerados influentes, com conteúdos originais, e com grande expertise em suas áreas de atuação a partir de seleção realizada pela revista Forbes[2]. Foram selecionados os vídeos mais relevantes de cada canal, postados entre 30 de março de 2007 e 15 de julho de 2019. Por fim, nesta base, além das transcrições de áudio e demais metadados dos vídeos (como, por exemplo: número de visualizações, curtidas,

[1]Todas as bases estão disponíveis para download no seguinte link: <https://www.ppgia.pucpr.br/~paraiso/Projects/YouGraph/>

[2]Forbes Top Influencers: <https://www.forbes.com/top-influencers/>

etc.), foram também extraídos os 100 comentários mais relevantes de cada vídeo. É importante ressaltar que o critério de "relevante" utilizado neste contexto refere-se ao algoritmo de seleção proprietário do Youtube, de forma que não é possível determinar quais métricas e dados são considerados na análise de relevância (mais detalhes na seção 3.1).

Já na base produzida a partir de vídeos em português brasileiro, foram extraídos cerca de 34 mil vídeos, distribuídos em 103 canais, compondo 26 categorias distintas (detalhes na tabela 6.2). Destes 34 mil vídeos, cerca de 95% (32.682 vídeos) foram obtidos através de um algoritmo ASR e 5% (1.881 vídeos) a partir de transcrição manual, conforme apresentado na figura 6.1. Como critério de escolha destes canais, foram selecionados os produtores de conteúdo vencedores do "Prêmio Influenciadores" dos anos 2019 e/ou 2018[3].

Categoria	**Total de Vídeos**	**Representatividade**
Beleza	1.352	12,10%
Games	1.293	11,57%
Entretenimento	1.265	11,32%
Moda	1.142	10,22%
Saúde	1.129	10,10%
Tecnologia e Negócios	914	8,18%
Home	906	8,11%
Família	874	7,82%
Comida	817	7,31%
Viagens	755	6,75%
Kids	730	6,53%
Total	**11.177**	**100%**

Tabela 6.1: Total de vídeos extraídos por categoria. Base Inglês

Em relação à base rotulada de tuplas de conhecimento, ela será apresentada na seção 6.3, pois, para a sua geração, foi necessário aplicar o extrator de informações aberto, que será discutido mais adiante. Por ora, vale mencionar que esta base possui 4.146 tuplas de conhecimento, extraídas de 314 sentenças a partir de 40 vídeos selecionados aleatoriamente da base *PT-BR*.

[3]O "Prêmio Influenciadores" é um evento mantido pela CECOM (Centro de Estudos da Comunicação - USP), que tem como objetivo encontrar os produtores de conteúdo mais influentes (relevantes) do Brasil, através de votações de uma equipe técnica e voto popular.

Categoria	Total Canais	Total Videos
Gastronomia	7	3.325
Tecnologia Digital	5	2.644
Games	5	2.641
Economia, Política e Atualidades	5	2.442
Comportamento e Estilo de Vida	5	2.208
Empreendedorismo e Negócios	6	2.202
Conhecimento e Curiosidades	4	1.937
Cultura e Entretenimento	4	1.915
Decoração, Organização e Diy	5	1.801
Esporte	4	1.764
Educação	4	1.716
Moda	5	1.592
Beleza	4	1.532
Pets	4	1.365
Fitness	5	1.315
Família	5	1.308
Meio Ambiente e Sustentabilidade	4	1.307
Saúde	4	1.022
Viagem e Turismo	5	893
Humor	4	817
Militar	1	577
Política	1	542
Cidades, Arquitetura e Urbanismo	4	530
Ciências	1	516
História	1	326
Programação	1	190

Tabela 6.2: Total de canais e vídeos extraídos por categoria. Base PT-BR

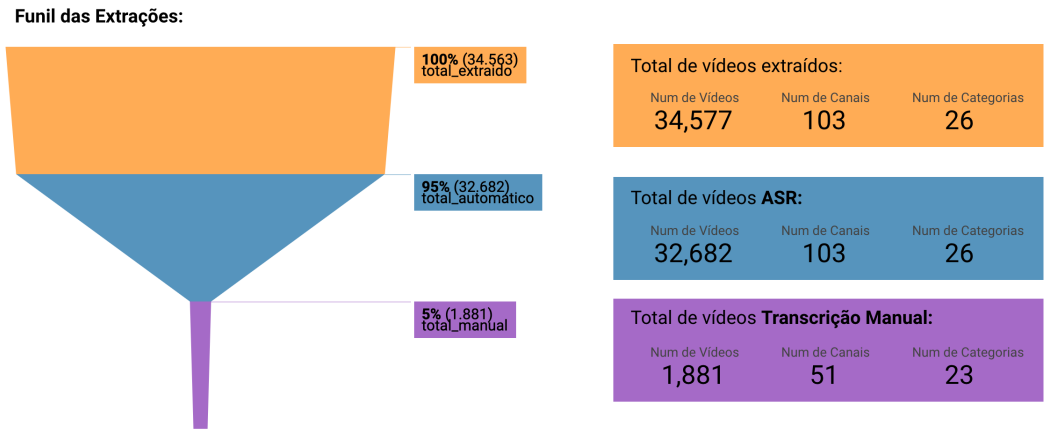


Figura 6.1: Total de vídeos obtidos via ASR e transcrição manual na base PT-BR

6.2 Um método de modelagem de tópicos em transcrições de vídeos do Youtube

Para a avaliação do método proposto na seção 5.4, utilizando o corpus de vídeos PT-BR, foram realizados experimentos com o objetivo de: (i) treinar diversos modelos de modelagem de tópicos para encontrar o melhor número de tópicos a partir de sua coerência; (ii) avaliar a representatividade dos tópicos do modelo selecionado na classificação dos documentos do corpus; e, (iii) avaliar o critério de seleção dinâmica de modelos proposto no trabalho de Rodrigues e Paraiso (2020).

Após realizar o pré-processamento dos dados, foi realizada a extração dos tópicos latentes presentes nas transcrições. Seguindo o método de critério de parada, foram gerados 64 modelos de modelagem de tópicos utilizando o algoritmo *LDA* com uma janela deslizante (jd) de 10 modelos (coeficiente de correlação = -0,198). A diferença entre cada modelo é o número de tópicos k gerados. A cada modelo gerado, o número k era incrementado em 1, começando no primeiro modelo com um $k = 26$ e o último modelo com um $k = 90$. Justificamos o inicio do treinamento com um k inicial igual a 26, pelo fato da base de dados possuir 26 categorias distintas. Ao final deste processo é calculada a coerência destes modelos para determinar qual seria o melhor modelo para ser utilizado nas classificações dos documentos. Devido ao fato do *LDA* possuir um caráter de inicialização aleatória, este processo foi executado três vezes e posteriormente foi calculada a média das coerências. Na Figura 6.2, podemos visualizar as coerências obtidas pelos modelos (linha azul), bem como os coeficientes de correlação gerados a partir das janelas deslizantes (barras vermelhas). Notem que no modelo $k = 90$, ocorre uma reversão no coeficiente de correlação demonstrando que o modelo parou de aprender.

Após a seleção do melhor modelo ($k = 80$), dois especialistas humanos analisaram os tópicos gerados a partir de suas palavras chaves e seus respectivos pesos. Após esta análise foi detectada a presença de diversos tópicos pouco representativos e majoritariamente compostos por palavras genéricas, além da presença de um tópico composto somente por palavras de baixo calão (tópico 56) e outro composto majoritariamente por nome próprios (tópico 60). A Tabela 6.3, mostra alguns destes exemplos. Desta forma, buscando uma melhor representatividade dos tópicos latentes e, seguindo o método apresentado, foi selecionado com modelo com um $k = 50$ como o melhor modelo, com uma coerência de 45,4%. Isto foi feito em função de ambos os modelos serem considerados similares, dado que possuem uma diferença de coerência menor ou igual a 2%.

Na figura 6.3a, é possível verificar a distribuição dos tópicos criados a partir de 4 categorias diferentes, utilizando o modelo selecionado ($k = 50$). Nesta figura, os números

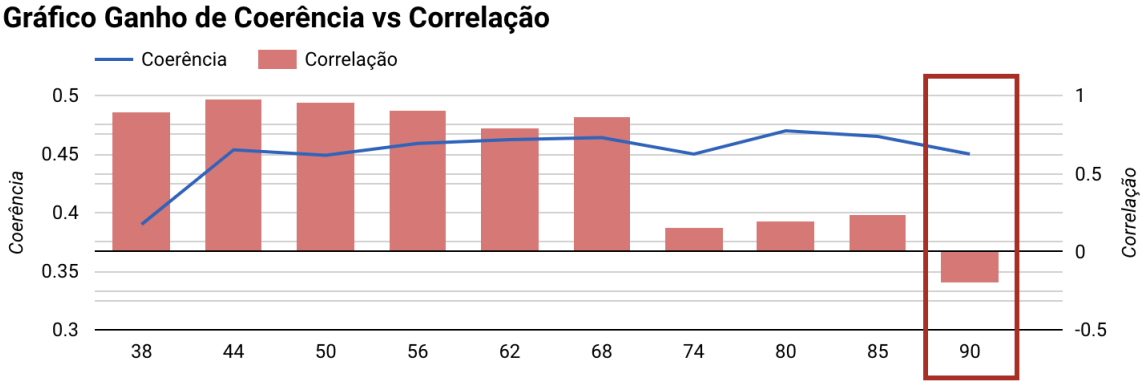


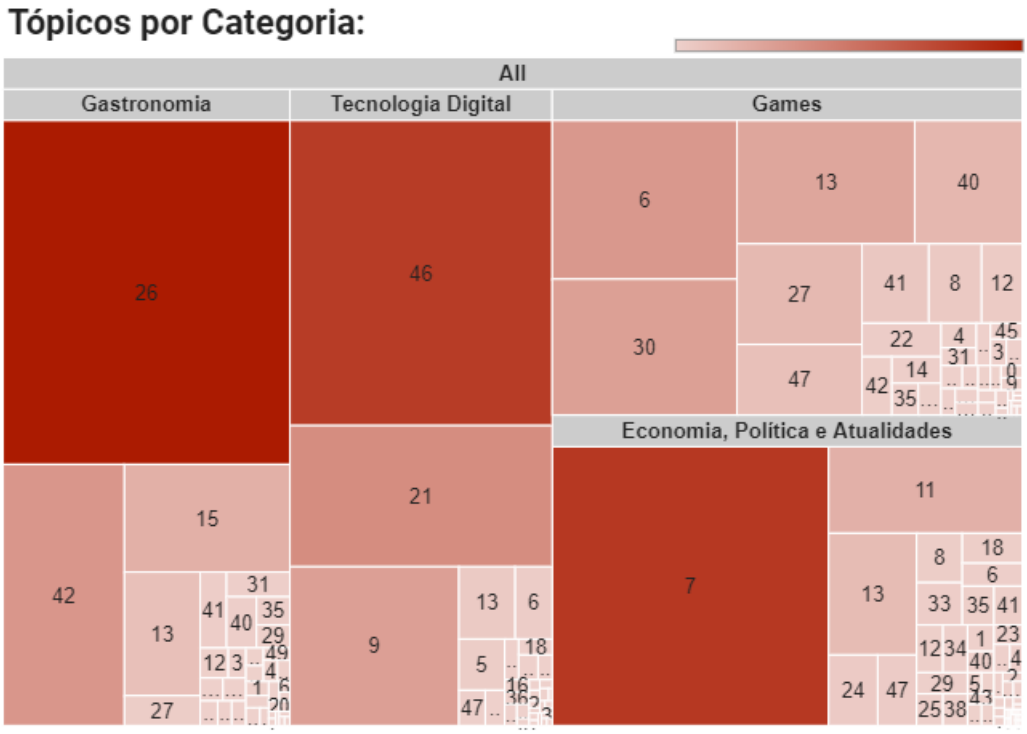
Figura 6.2: Distribuição dos tópicos criados pelos canais

Id do Tópico	Palavras
46	ruim, amigar, certeza, certar, problema, absurdo, melhor, época, sacar, causar, parar
54	piscina, rápido, clicar, último, show, pessoal, lugar, próximo, louco, rodar, comentário
56	po*ra, ca***ho, m***a, fo**r, pu*a, velho, moleque, bo**a, bagulho, ruir, rapaz
60	abraçar, mensagem, mateus, josé, silvar, novo, noite, carlos, fábio, filhar, maria
69	futurar, dezembro, nome, maio, comedir, julho, casar, início, junho, vidar, voltar

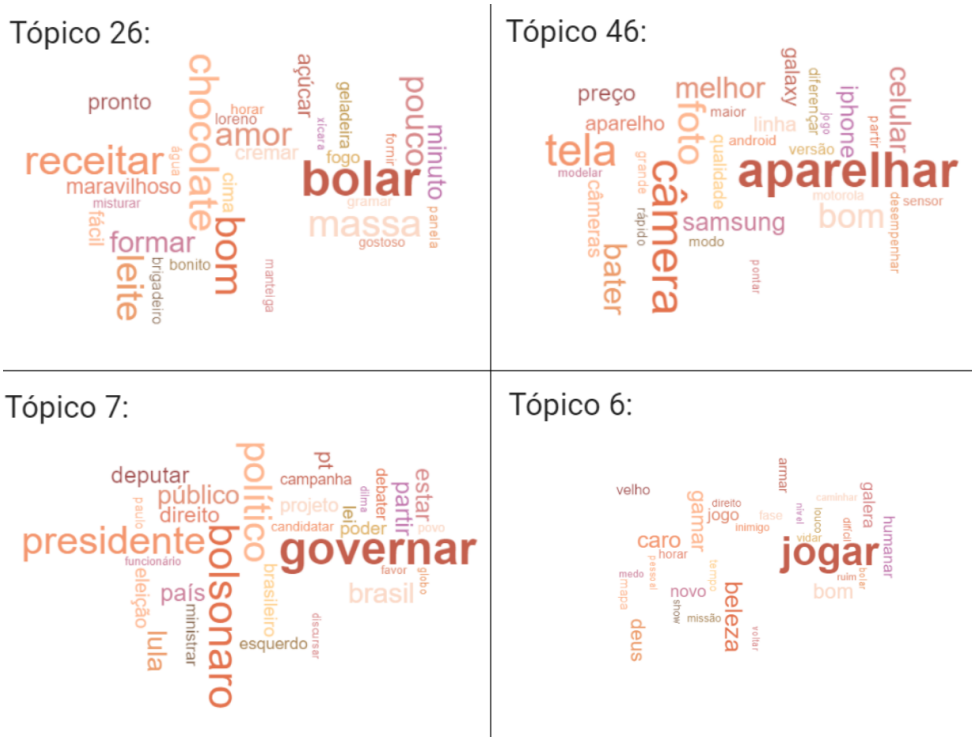
Tabela 6.3: Exemplos de tópicos pouco representativos no modelo de k=80.

dentro dos quadros representam os identificadores únicos de cada tópico, e quanto mais forte a sua cor e maior o seu tamanho, maior é a quantidade de vídeos classificados dentro deste tópico. Já na figura 6.3b é apresentada a composição dos principais tópicos do modelo a partir de seu *id*. Vale mencionar que, os tamanho de uma palavra nesta nuvem de palavras significa o peso, ou relevância, destes termos dentro do tópico analisado. Desta forma, é possível compreender os significados dos tópicos a partir das palavras chaves que os compõe. Confrontando ambas as figuras (6.3a e 6.3b) é possível notar que os resultados apresentados pela modelagem de tópicos são condizentes com as categorias dos canais.

Indo além, explorando a representatividade destes tópicos em (MUNARO et al., 2021a), dois avaliadores humanos especialistas na área de marketing foram incumbidos de atribuir um conjunto de rótulos aos grupos gerados pelo modelo obtido. Assim sendo, os 50 tópicos obtidos através do algoritmo *LDA*, geraram 19 grupos nomeados conforme



(a) Distribuição dos tópicos criados pelos canais



(b) Palavras mais representativas

Figura 6.3: Tree Map e Wordcloud gerado pelo LDA

apresentado na Tabela 6.4.

Destes 19 grupos, os 5 tópicos mais representativos representam 26% de todos

os vídeos coletados. O *tópico 13* teve o maior número de vídeos relacionados a ele, aparecendo em 95 canais diferentes. Já o *tópico 26* apareceu em 43 canais diferentes, e conforme apresentado na Figura 6.3b, suas palavras-chave remetem a termos culinários e de gastronomia em geral. O *tópico 7* foi discutido em 31 canais diferentes, e seus conteúdos remetem a palavras do contexto político (ver figura 6.3b). Em seguida, o *tópico 0*, que aparece em 43 canais, aborda assuntos sobre estética e maquiagem. Por fim, o *tópico 46* trata sobre temas referentes ao cenário de tecnologias, como por exemplo celulares e computadores.

Ao confrontarmos as informações presentes nas tabelas 6.2 e 6.4, fica evidente que os conteúdos produzidos pelos influenciadores digitais não se restringem as suas categorias de origem. Por exemplo, na base utilizada, somente sete canais se intitulavam como um canal de gastronomia. No modelo gerado, um dos tópicos equivalentes a este domínio é o *tópico 26*, que está presente 43 canais diferentes. Na Tabela 6.5 estão representados os 5 principais tópicos criados, o número de canais que se intitulam a este domínio e o número de canais onde foram encontrados vídeos que abordam estes tópicos.

Finalmente os resultados destas análises mostraram que as categorias de "Educação", "Cultura e Entretenimento", "Pessoas, Comportamento e Estilo de Vida" e "Gastronomia", são os assuntos mais populares entre os influenciadores digitais dentro do Youtube. "Educação" é o assunto mais popular na base de vídeos analisada. 3.555 vídeos são majoritariamente compostos com tópicos referentes a este domínio, representando 10,3% de toda a base. Este resultado é consistente com um estudo do Google, onde é demonstrado que uma das principais razões que as pessoas acessam o Youtube é para buscar novos conhecimentos e aprendizados (GOOGLE, 2019).

6.3 Um método para extração e refinamento de tuplas de conhecimento

Após a obtenção dos tópicos, iniciou-se a extração das tuplas de conhecimento utilizando o método proposto por (SENA; CLARO, 2018). Conforme explicado na seção 5.7, as tuplas possuem um formato de *tupla = (argumento, relação, argumento)*. Ao todo foram extraídas mais de 23 mil tuplas de conhecimento de 51 vídeos escolhidos de forma aleatória do corpus *PT-BR* a partir de transcrições realizadas manualmente. É importante salientar que na geração do espaço amostral foram selecionados somente vídeos com transcrições obtidas de forma manual, pois o extrator de informação aberto é incapaz de extrair fatos de sentenças sem pontuações (mais detalhes sobre isto no capítulo

7).

A tabela 6.6 demostra alguns exemplos das tuplas geradas a partir de um trecho de transcrição[4]:

O michael jordan ganhou o título da nba em 91 92 93 96 97 98 ele tem tanto título, que não dá para ficar falando tudo em um vídeo só, é muita coisa. Esse cara talvez tenha sido o melhor jogador de basquete todos os tempos e talvez um dos melhores, senão o melhor esportista de todas as modalidades de todos os esports da história do nosso planeta.

Analisando a tabela apresentada, é possível notar que o extrator de informações extraiu corretamente dois fatos relevantes: (i) de que o jogador de basquete Michael Jordan ganhou diversos títulos da *NBA* e (ii) de que talvez ele tenha sido o melhor jogador da história.

Na tabela 6.7 podemos verificar mais alguns exemplos de fatos que podem ser extraídos a partir de um extrator de informações aberto. Um caso interessante é a tupla de número 2 do exemplo, onde o sistema cita corretamente que o carnaval de 2016 do rio teve um público de pouco mais de 18 mil pessoas. Também podemos notar a presença de alguns ruídos nas extrações, como no exemplo de número 3, onde apesar da extração estar correta, o sistema acabou adicionando o termo "mil" no argumento 2.

Analisando ambas as tabelas apresentadas, podemos concluir que os extratores de informações abertos constituem uma importante ferramenta na recuperação de informações em transcrições de vídeos. Entretanto, conforme já explicado, os extratores *OpenIe*, possuem algumas limitações que podem ser potencializadas ao serem utilizados em transcrições de vídeos da Internet, gerando extrações errôneas. Entre os principais erros que podem acontecer, podemos citar:

- **1 - Erros de transcrição:** Transcrições de áudio extraídos a partir de algoritmos de *ASR* podem conter erros de transcrições. Consequentemente, a extração de um fato de um texto incorreto automaticamente também estará incorreta. A tabela 6.9 apresenta alguns desses erros.

- **2 - Relações desconexas, repetidas e similares:** para melhor discutirmos esta questão, selecionamos um vídeo aleatório da base (*COMO VOU EMAGRECER 23KG EM 4 MESES! (E a relação entre dieta e investimentos!)* do canal *O Primo Rico* [5]). Este vídeo em específico possui um total de 3.221 tuplas de conhecimento.

[4]Fonte do vídeo: <https://www.youtube.com/watch?v=16lr-G4s9KY>
[5]Link: https://$www.youtube.com/watch?v = 7U_2g1POsVI$

A tabela 6.8 demostra algumas tuplas extraídas. Neste exemplo podemos constatar duas coisas: 1º, essas tuplas não possuem valor semântico. 2º, As tuplas são extremamente similares entre si. Este comportamento de tuplas similares e com informações sem valor semântico se repete pelos outros vídeos. Uma possível estratégia para mitigar estes problemas seria aplicar algum algoritmo de similaridade entre as tuplas e utilizar algum classificador para classificar a semântica da tupla. Por fim, também foi detectada uma enorme quantidade de triplas repetidas na base. Das 3.221 extrações realizadas no vídeo citado, somente 240 são únicas. A hipótese que levantamos para esta situação é devido aos problemas de pontuação que discutiremos no tópico a seguir. Como o algoritmo não consegue detectar corretamente as árvores de dependência, o sistema acaba repetindo os dados, gerando a duplicidade das tuplas.

- **3 - Problema nas pontuações:** Para tentar solucionar o problema da falta de pontuação das transcrições de áudio, tentamos utilizar a ferramenta proposta em López e Pardo (2015), com o objetivo de adicionar pontuações ao nosso corpus de forma automática. Esta ferramenta foi escolhida por possuir bons resultados na literatura. Infelizmente não conseguimos nenhum resultado satisfatório com o corpus utilizado nesta pesquisa. Trabalhamos com a hipótese de que textos de transcrições podem possuir uma estrutura gramatical diferente do corpus utilizado em (LÓPEZ; PARDO, 2015), que trabalhou com textos jornalísticos.

É importante salientar que o tratamento dos supracitados problemas, não faz parte do escopo deste trabalho. Contudo, a fim de refinar a obtenção de informações oriundas do processo de transcrição, foi também proposto o desenvolvimento de uma abordagem com o objetivo de tratar as tuplas repetidas ou muito similares. Esta priorização justifica-se principalmente pelo impacto deste tipo de erro nas extrações realizadas.

Assim sendo, todas as tuplas extraídas a partir do extrator de (SENA; CLARO, 2018), foram "refinadas" seguindo o método apresentado na seção 5.8. Vale relembrar que o processo do refinamento das tuplas de conhecimento possui três objetivos principais: (i) encontrar grupos de tuplas similares a partir de uma sentença de entrada, (ii) inferir quantas tuplas melhor define uma sentença (chamado de *n° de tuplas candidatas*) e, finalmente, (iii) encontrar a tupla mais representativa a partir de um dado grupo.

Entretanto, para que seja possível avaliar o refinamento das tuplas, é necessário a existência de uma base de comparação, com informações corretas e confiáveis destes agrupamentos e refinamentos (conforme discutido na seção 4.4.4).

Para este fim, foi gerado um espaço amostral estratificado proporcional, onde três

avaliadores humanos com *expertise* na área de processamento de linguagem natural e *marketing* rotularam um conjunto de tuplas determinando os grupos de tuplas mais similares entre si[6]. Os extratos foram definidos da seguinte forma: foram criadas subpopulações a partir do número de tuplas extraídas em cada sentença, agrupadas de cinco em cinco. Na figura 6.4 é possível notar que os extratos seguem uma distribuição exponencial, com a grande maioria das sentenças possuindo menos de 5 tuplas extraídas.

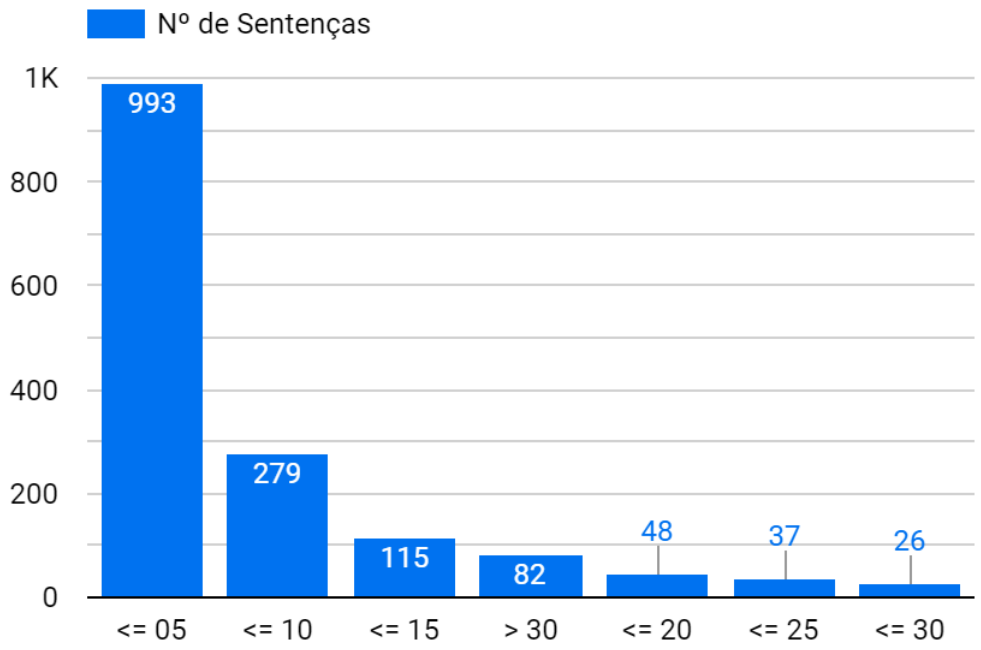


Figura 6.4: Total de sentenças distribuídas por extratos

Seguindo a distribuição dos extratos, foi gerado um espaço amostral de 314 sentenças (nível de confiança de 95% e 5% de erro amostral) compondo 4146 tuplas para serem avaliadas manualmente, de forma independente pelos avaliadores humanos.

Uma vez finalizado os agrupamentos manuais, objetivando determinar se os avaliadores estavam ou não concordantes entre si, foi realizado o teste estatístico *ICC (Intraclass Correlation Coefficient)* para determinar o coeficiente de concordância r entre eles. Obteve-se um $r = 0.942$, demostrando que os avaliadores tiveram uma alta concordância em suas avaliações.

De posse das três bases rotuladas pelos avaliadores, foi executada uma etapa de junção dos grupos obtidos pelos avaliadores a fim de obter uma representação única (um único conjunto de grupos), condizente com os conjuntos de grupos individuais indicados por cada avaliador. O conjunto único de grupos resultante deste processo é aqui

[6]Para isto, foi desenvolvida uma ferramenta própria para este fim, utilizando a plataforma *Google Sheets*

denominado "padrão-ouro".

A Figura 6.5 ilustra o processo de obtenção do padrão-ouro a partir dos agrupamentos individuais resultantes de cada avaliador. O conjunto único de grupos (padrão-ouro) representa o melhor consenso entre os conjuntos de grupos de entrada.

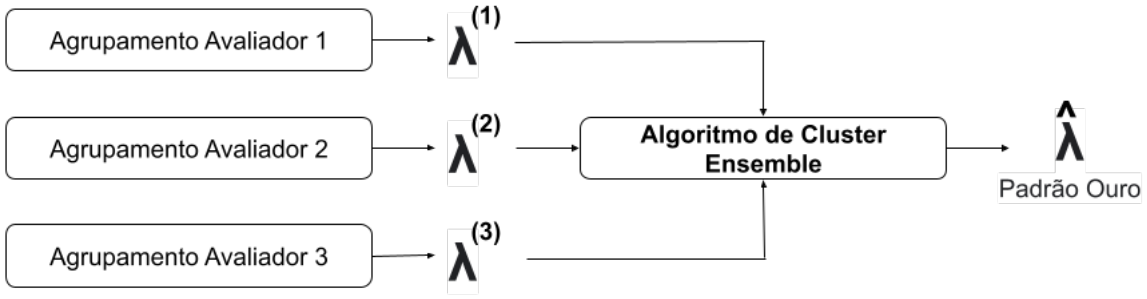


Figura 6.5: Obtenção do padrão ouro a partir dos agrupamentos individuais.

Para fazer este agrupamento de agrupamentos (*Cluster Ensembles*) foi utilizado o algoritmo *MCLA (Meta-CLustering Algorithm)*, que realiza um mapeamento *n para n* de cada grupo criado por cada avaliador, com todos os grupos dos outros avaliadores. Todo este mapeamento é representado a partir de um *Hipergrafo* (STREHL; GHOSH, 2002). Realizada esta etapa, novamente foi executado o teste *ICC* para determinar se este novo agrupamento estava concordante com os realizados pelos avaliadores humanos. Foi obtido um $r = 0.873$, significando que o padrão ouro gerado é concordante com os avaliadores humanos.

Uma vez gerado o padrão-ouro, a próxima etapa na execução do refinamento das tuplas de conhecimento consiste em executar os agrupamentos das tuplas similares a partir dos métodos descritos na seção 5.8.

A tabela 6.10 apresenta todos os métodos de agrupamento de tuplas similares executados, juntamente com o percentual de falha no processo de agrupamento (realizada em todas as sentenças do corpus). Entende-se como falha, a incapacidade do método conseguir inferir o número de tuplas candidatas em uma dada sentença. Por legibilidade, a cada experimento realizado, foi atribuído um nome (chamado de *tag método*).

Os 6 primeiros experimentos apresentados na tabela são referentes à técnica de agrupamento a partir de cálculos vetoriais (similaridade do cosseno). O cálculo das similaridades do cosseno foi feito com 3 representações vetoriais diferentes (*Bert*, *Tf-Idf* e *Word2Vec*) e 2 limiares de similaridade diferentes: 70% e 80%. Dos métodos baseados em similaridade do cosseno, o que teve um menor percentual de falha foi o *word2vec-70* (com 7,22%) seguido pelo *bert-70* (com 8,10%).

Os experimentos 7 e 8 são métodos baseados em aprendizagem não supervisionada (a partir do algoritmo *K-Means*). Neste método foi utilizado duas representações vetoriais diferentes: a partir do *Word2Vec* (tag método *embedding_distortion*) e a partir do *Word2Vec* aplicando o algoritmo redução de dimensionalidade *T-SNE* (tag método *tsne_distortion*)[7]. Para determinar os melhores parâmetros do modelo *T-SNE*, foram gerados 10 modelos diferentes alterando incrementalmente o valor da perplexidade. O treinamento dos modelos foram iniciados com uma perplexidade igual a 5, sendo este valor incrementado em 5 até um valor máximo de 50. Para facilitar o ajuste dos hiperparâmetros, definimos uma taxa de aprendizado como automática.

Analisando os resultados podemos verificar que a tag método *embedding_distortion* foi o método que teve o pior resultado, sendo incapaz de inferir o número de tuplas candidatas em mais de 58% das sentenças. A seguir temos o *tsne_distortion* como o segundo pior colocado nos experimentos com cerca de 50% das sentenças com falha na extração. Vale mencionar que o algoritmo do *T-SNE* se mostrou uma interessante ferramenta para melhorar a performance do agrupamento de tuplas similares, garantindo uma melhora de aproximadamente 8% nas extrações.

Entretanto, ainda assim, os modelos gerados a partir de algoritmos de aprendizagem de máquina não se provaram uma solução viável para a determinação do número de tuplas candidatas. Podemos justificar esta baixa performance analisando a figura 6.6. Nesta imagem, são apresentados dois gráficos para os tag métodos baseados em aprendizagem não supervisionada (*tsne_distortion* e *embedding_distortion*). Neste gráfico podemos visualizar três informações relevantes: o número de sentenças onde foi possível inferir o número de tuplas candidatas, o total de sentenças onde houve falha, e finalmente o percentual de falhas acumulado pelo total de falhas obtido no método em questão. Todas as informações estão relacionadas aos extratos obtidos.

Pode-se notar que mais de 90% de todas as falhas estão reunidas nos extratos referentes a menos de 10 tuplas por sentença. Podemos justificar este fato como principal fator limitante ao método *K-Means* para a geração de bons resultados. Como a maioria das sentenças gera um baixo número de tuplas, o modelo é incapaz de generalizar os conjuntos de dados de entrada, e, consequentemente, é incapaz de inferir um número ideal de grupos para representar a sentença de entrada.

Com isso, o método de refinamento de tuplas de conhecimento aqui apresentado, após este processo, baseia-se apenas nos Tag Métodos baseados em distâncias do cosseno,

[7] O algoritmo *T-SNE* não foi originalmente desenvolvido com o intuito de servir como uma técnica de redução de dimensionalidade, entretanto diversos estudos mostram que o *T-SNE* pode ser técnica interessante para este fim.

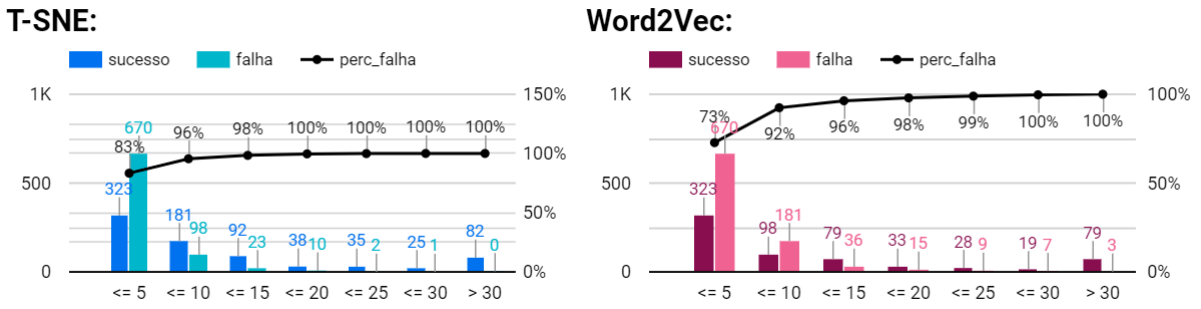


Figura 6.6: Percentuais de falhas dos tag métodos *embedding_distortion* e *tsne_distortion*.

que melhor generalizam as tuplas de entrada em um conjunto de grupos representativos.

Neste sentido, o próximo passo é analisar a qualidade dos grupos gerados. Isto é feito aplicando o algoritmo de concordância de *Jaccard* apresentado na seção 5.8. Este algoritmo recebe como entrada dois agrupamentos distintos (aqui, os conjuntos de grupo "padrão-ouro" e os obtidos pelos "Tag Método"), e calcula a similaridade entre estes grupos a partir do total de grupos mapeados corretamente a partir do total de grupos do primeiro conjunto (neste caso, do padrão-ouro).

A Tabela 6.11 mostra a porcentagem de concordância obtida entre o padrão-ouro e os Tag Métodos. Os resultados apresentados consideram a concordância de *Jaccard* com um limiar de similaridade entre grupos de 100%.

Para verificar possíveis vantagens na utilização exclusiva de um Tag Método específico em relação aos demais, este mesmo teste de similaridade entre grupos foi executado considerando também limiares de similaridade a partir de 75% com intervalos de 5% até a similaridade completa.

A tabela 6.12 mostra os coeficientes de similaridade entre os Tag Métodos considerando os limiares de similaridade de 75% até 100%.

O Tag Método bert-cosseno-70 apresentou os melhores coeficientes de similaridade entre grupos, independentemente do limiar utilizado, seguido pelos Tag Métodos bert-cosseno-80 e word2vec-80.

Os coeficientes de similaridade obtidos pelos diferentes Tag Métodos foram submetidos ao Teste de Friedman (FRIEDMAN, 1937), teste estatístico não-paramétrico para comparação de múltiplas amostras, conforme protocolo descrito em (DEMŠAR, 2006), utilizando nível de significância de 95%, e considerando a hipótese nula de igualdade entre os Tag Métodos.

O Teste de Friedman revelou diferença significativa entre as amostras ($p\text{-}value = 0.0000$) e permitiu a realização de um teste *post-hoc* a fim de identificar diferenças entre

os pares de Tag Método. Para tal análise foi utilizado o teste de Nemenyi (NEMENYI, 1962), que identifica diferenças entre duas amostras quando suas posições médias diferem mais que um determinado valor crítico (CD, do inglês *Critical Difference*).

A Figura 6.7 mostra o gráfico de diferenças críticas (CD = 3,0780) obtido na comparação entre os coeficientes de similaridade obtidos pelos diferentes Tag Método. Como observado, apesar dos melhores resultados obtidos pelo Tag Método (descritos na Tabela 6.12), o Teste de Nemenyi não revelou diferença significativa entre os Tag Métodos bert-cosseno-70, word2vec-80, bert-cosseno-80 e word2vec-70.

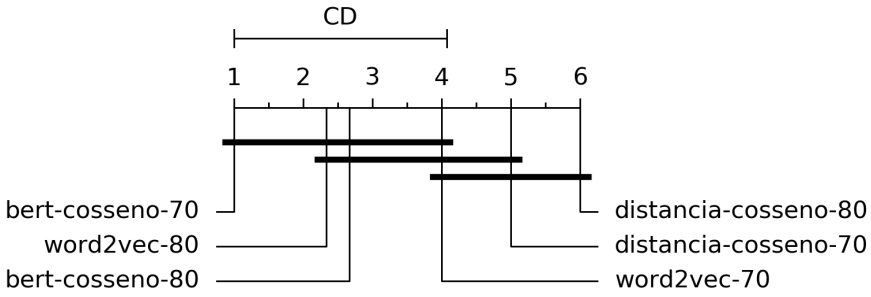


Figura 6.7: Gráfico de diferenças críticas para coeficientes de similaridade obtidos pelos diferentes Tag Métodos.

Com isso, o agrupamento de tuplas obtido pelo Tag Método bert-cosseno-70 foi utilizado para continuidade do processo de refinamento de tuplas e para obtenção da tupla mais representativa de seus grupos componentes. Apesar dos Tag Métodos com melhores resultados não apresentarem diferença significativa entre si, o Tag Método bert-cosseno-70 ainda sim obteve os melhores coeficientes de similaridade independentemente do limiar observado (conforme Tabela 6.12).

6.4 YouGraph e suas aplicações práticas

Uma vez gerado um grafo de conhecimento utilizando os métodos propostos e o *framework* desenvolvido, foram realizadas pesquisas paralelas ao projeto, que se valeram dos benefícios do uso de redes e grafos no seu desenvolvimento. Desta forma, estes trabalhos são aplicações diretas dos resultados obtidos com o método apresentado. Estas pesquisas podem ser resumidas nos seguintes projetos:

- Um método para identificação de conteúdo patrocinado em vídeos do Youtube;
- Uma análise das principais características que tornam um vídeo influente (viral);

6.4.1 Um método para identificação de conteúdo patrocinado em vídeos do Youtube

Neste trabalho, foi explorado uma nova técnica para a detecção automática de conteúdo patrocinado em vídeos do Youtube, a partir da extração de entidades nomeadas (marcas patrocinadas) e seus eventuais relacionamentos nas tuplas de conhecimento obtidas a partir de um extrator de informações. Este estudo possui grande relevância para a sociedade e comunidade acadêmica, dado o contexto atual dos influenciadores digitais na disseminação de novos conteúdos (GOEL; WATTS; GOLDSTEIN, 2012), influência social (AGARWAL et al., 2008) e na formação de opiniões coletivas (KWON; HAN; KIM, 2017). Desta forma, alguns autores já tentaram abordar o tema da detecção automática de patrocínio em vídeos, como por exemplo o trabalho de Gerhards (2019), que conduziu um estudo exploratório manual de marcas patrocinadas a partir de 27 canais do Youtube, ou em (SCHWEMMER; ZIEWIECKI, 2018), onde os autores utilizaram uma abordagem baseada em modelagem de tópicos para inferir se um dado vídeo possuía um conteúdo de alguma marca e/ou produto, a partir de uma base de 100 canais diferentes. No entanto, este método ainda era incapaz de detectar a presença real de conteúdo patrocinado, limitando-se em determinar se um dado vídeo apresentava ou não algum produto. Conforme apresentado por Alexe et al. (2012), diferentes redes sociais possuem diferentes características que podem tornar o seu uso mais desafiador. No caso das transcrições do Youtube, a falta de pontuação na maioria dos vídeos pode limitar o desenvolvimento de ferramentas de detecção de conteúdos patrocinados.

Desta forma, a maior contribuição do método que será apresentado nesta seção é a flexibilidade da ferramenta na detecção de propagandas, seja em vídeos com pontuações (transcrição manual) ou sem pontuações (transcrição *ASR*). Na figura 6.8 é apresentado o método responsável pela detecção de conteúdos patrocinados, que é composto por seis etapas principais: (i) aquisição dos dados, (ii) pré-processamento dos documentos, (iii) identificação dos vídeos candidatos, (iv) processamento das transcrições manuais, (v) processamento das transcrições automáticas e (vi) filtragem das tuplas candidatas. Vale mencionar que os processos da aquisição dos dados já foram apresentados na seção 4.2 e o pré-processamento na seção 5.1.

Desta forma, adiante serão apresentadas as demais etapas necessárias para realizar a classificação automática de conteúdo patrocinado. Para isto, é importante o entendimento de três conceitos chave, que serão utilizados no discorrer deste trabalho: *"vídeos candidatos"*, *"sentenças candidatas"* e *"tuplas candidatas"*. O termo candidato remete ao fato de um documento (vídeo, sentença ou tupla) possuir uma alta probabilidade de seu

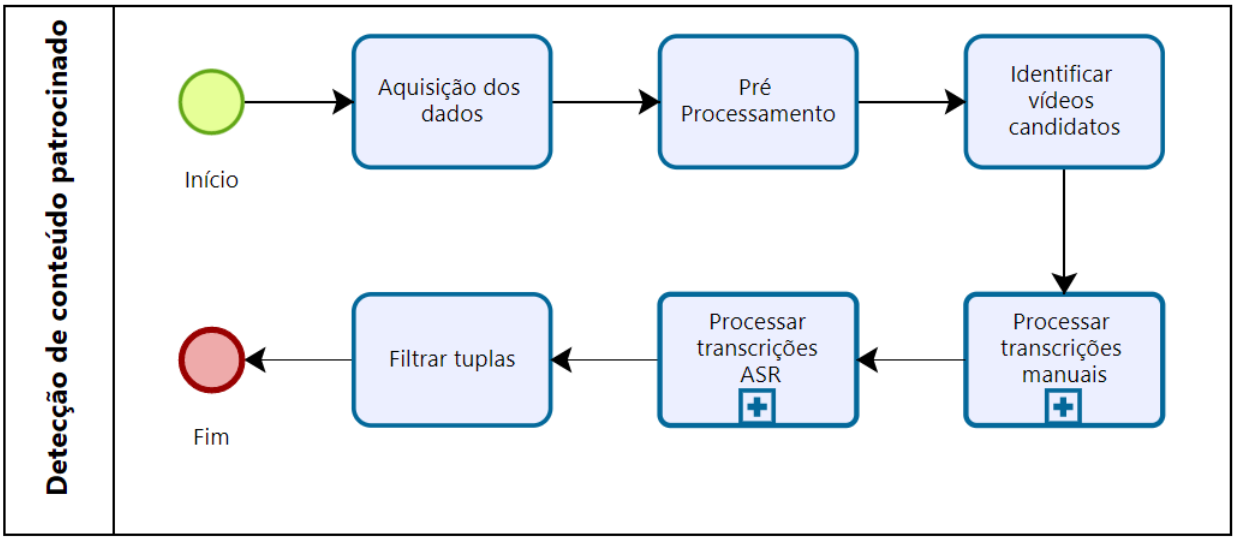


Figura 6.8: Método proposto para extração de conteúdo patrocinado.

conteúdo ser patrocinado, que ainda não foi confirmado pelo método proposto.

Indo adiante, após a realização da coleta e pré-processamento dos dados, o próximo passo a ser executado é a identificação dos "vídeos candidatos". Isto é feito realizando uma busca por um conjunto de palavras que remetam ao contexto de patrocínio. Estas palavras chaves foram selecionadas a partir de uma análise de dois especialistas na área de marketing (apresentado no anexo 8.3).

Em seguida, é executado o "processamento das transcrições manuais"(ver Figura 6.9). Isto é feito, selecionando todos os vídeos candidatos obtidos com transcrições manuais e, destes vídeos, são extraídas e armazenadas todas as sentenças que continham as palavras-chave para posterior processamento. O processo de segmentação das sentenças manuais foi apresentado na seção 5.6. A seguir, foram identificadas todas as sentenças que continham uma ou mais entidades nomeadas, seguindo o método apresentado na seção 5.5. Assim sendo, caso uma sentença possua uma entidade nomeada e ainda possua uma das palavras-chave selecionadas, esta sentença passa a ser considerada candidata e continuará no processo de execução.

Após a obtenção das sentenças candidatas, todas as sentenças passarão no extrator de informação para serem geradas as suas respectivas tuplas de conhecimento (seção 5.7). Todas as tuplas extraídas são consideradas tuplas candidatas. Para efetivamente determinar se uma tupla é ou não patrocinada e consequentemente classificar um vídeo como patrocinado, é necessário realizar uma filtragem destas tuplas a partir de dois critérios:

- (i) a tupla possui uma ou mais palavras-chave?
- (ii) a tupla possui uma NER?

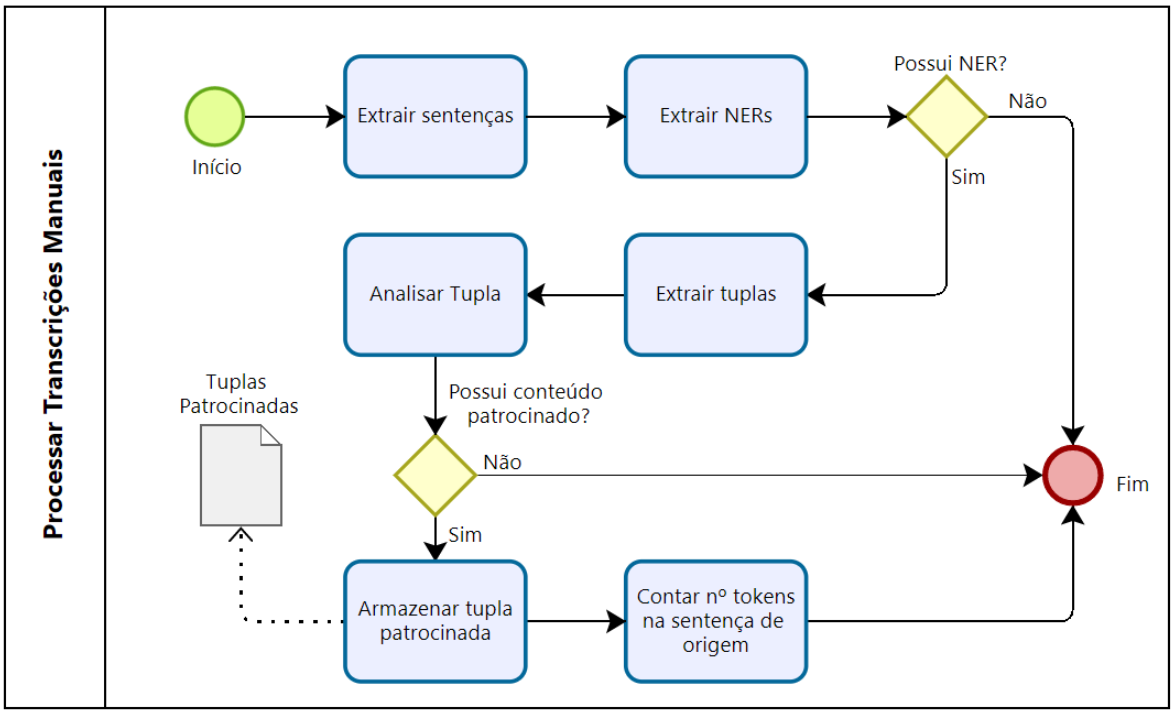


Figura 6.9: Processamento das transcrições manuais.

Todas as tuplas que atenderem estes requisitos serão consideradas patrocinadas.

Juntamente com a extração dos fatos, dois especialistas humanos realizaram uma análise manual nas sentenças candidatas para confirmar a presença ou não de conteúdo patrocinado. Esta verificação possui duas finalidades importantes: primeiro determinar a acurácia do método e segundo, garantir que o método é robusto na inferência do conteúdo patrocinado. Em seguida, será calculada a média de tokens das sentenças patrocinadas. Esta média será utilizada na geração das pseudo-sentenças das transcrições automáticas (*ASR*), conforme apresentado na Figura 6.10. Vale mencionar, que geração das pseudo-sentenças apresentadas neste trabalho, é uma adaptação do método discutido na seção 5.6. Dada a ausência de pontuações em transcrições *ASR*, é impossível segmentar as sentenças e consequentemente determinar a presença de uma marca patrocinada. A média de tokens obtidas nas transcrições manuais serve como uma solução alternativa para montar as *pseudo-sentenças* que serão utilizadas no extrator de informações. A *pseudo-sentença* é gerada da seguinte forma: Uma vez identificada uma palavra-chave de interesse, o valor da média de tokens é divido por dois para contemplar todas as palavras à esquerda e à direita da palavra-chave. A Figura 6.11 mostra um exemplo hipotético de uma média de 10 tokens e como ficaria a composição de uma *pseudo-sentença*. Por fim, após a obtenção das *pseudo-sentenças*, o fluxo para a identificação de conteúdo funciona de forma análoga ao processo da extração manual.

Para avaliar o método apresentado, foi utilizado a base PT-BR de cerca de 34 mil

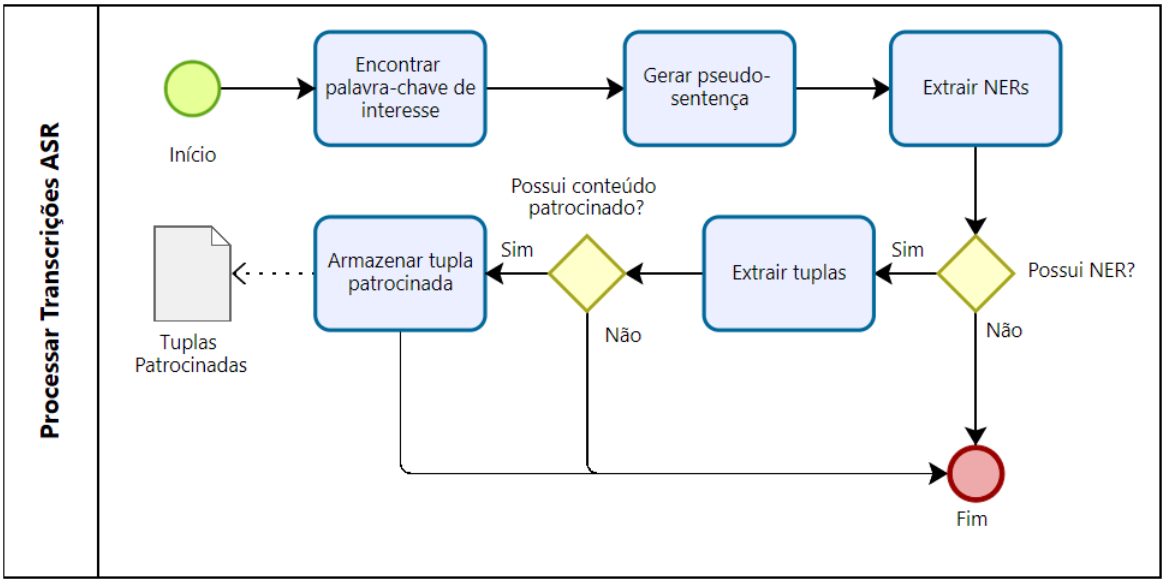


Figura 6.10: Processamento das transcrições automáticas.

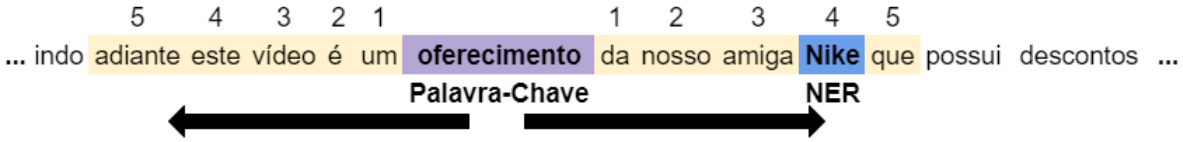


Figura 6.11: Exemplo da obtenção das *pseudo-sentenças*.

vídeos. Utilizando a lista de palavras-chave apresentada no anexo 8.3, foram encontrados 3.342 (9,66%) vídeos candidatos a patrocínio sendo 87 vídeos de transcrição manual e 3.255 *ASR*.

Aplicando o segmentador de sentenças nos vídeos manuais, foram obtidas 203 sentenças distintas. A análise humana (realizada por dois especialistas em *marketing* e PLN) determinou que 25,12% ($n = 51$) das sentenças efetivamente possuíam um caráter patrocinado de alguma marca. Nesta análise foi calculado a média de *tokens* das sentenças que ficou com um $x = 32$.

Já no processamento das transcrições *ASR*, os 3.255 vídeos candidatos geraram 4.767 *pseudo-sentenças*, com 6.137 entidades nomeadas detectadas. Na Tabela 6.13, é possível visualizar alguns exemplos das sentenças e pseudo-sentenças obtidas. Na Tabela 6.13 temos alguns exemplos das sentenças manuais e ASR.

Indo adiante, extraindo as tuplas de conhecimento das sentenças e pseudo-sentenças, foram extraídas 9.211 tuplas. Conforme já discutido anteriormente, as tuplas obtidas a partir de extratores de informações abertos, possuem como desvantagem o excesso de tuplas muito similares, processo que pode comprometer futuras análises e processamentos. Desta forma, foi aplicado a etapa de filtragem para se obter somente as tuplas que reme-

tessem ao contexto de vídeos patrocinados (com a presença de uma NER e palavra-chave). Foram obtidas 2.733 tuplas candidatas neste processo. Por fim, caso um vídeo analisado possua ao menos uma tupla candidata, este vídeo é considerado patrocinado. Ao término de todo o método apresentado foram obtidos 546 vídeos patrocinados, sendo 503 obtidos de transcrições automáticas e 51 manuais, conforme apresentado na Figura 6.12.

Na Tabela 6.14 , é apresentado um exemplo de um conjunto de tuplas extraído a partir de um vídeo com transcrição manual. Este exemplo chama atenção pelo fato do método ter classificado corretamente o vídeo como patrocinado, mesmo sem a referência direta ao canal de origem do vídeo (realizada a partir de uma citação do advérbio "aqui") (tupla número 4). Já as Tabelas 6.15 e 6.16 mostram dois exemplos do conjunto de tuplas gerados a partir de transcrições *ASR*. Em ambos os casos o método classificou corretamente o vídeo como patrocinado. Outro ponto que chama a atenção nas transcrições *ASR* é o fato das marcas e empresas citadas aparecerem em minúsculo. É interessante notar que este fato poderia confundir o modelo e impossibilitá-lo de compreender estas entidades nomeadas.

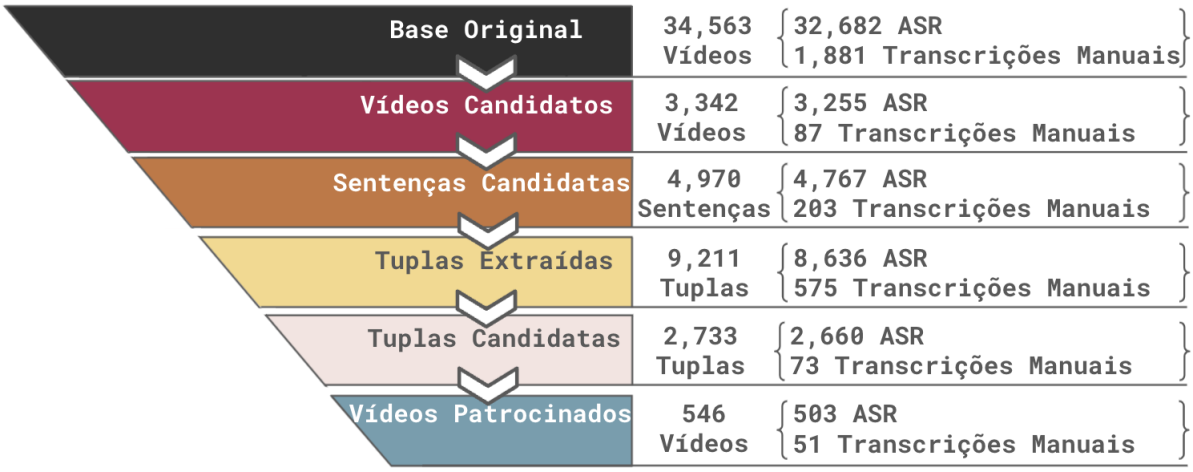


Figura 6.12: Resultados obtidos com a execução do método de detecção de vídeos patrocinados.

6.4.2 Uma análise das principais características que tornam um vídeo influente (viral)

O trabalho apresentado em Munaro et al. (2020) teve como objetivo identificar as principais características que determinam a popularidade de um vídeo em relação ao número de comentários (que demonstra haver uma discussão ao tema postado) e ao número de *likes/dislikes* (que por sua vez demonstra a aceitação do tema tratado no vídeo).

Neste trabalho, foi utilizada a base *EN-US* descrita na seção 6.1, composta por cerca de 11 mil vídeos, para gerar um modelo probabilístico com o objetivo de mapear quais características são mais correlacionadas na engajamento do público e consequentemente que potencializem a geração de *likes/dislikes* e comentários em um vídeo. Também foi gerado, um modelo de regressão para determinar a influência destas variáveis identificadas no modelo.

A Tabela 6.17 demostra as variáveis contínuas empregadas e a Tabela 6.18 demostras as variáveis categóricas. Ambas as tabelas contam com as estatísticas descritivas das variáveis.

Uma vez que os dados extraídos possuem características não paramétricas, não seria possível utilizar um modelo tradicional para a regressão dos dados (como por exemplo a regressão múltipla linear). Desta forma optou-se pelo uso da regressão binomial não negativa (HUGHES; SWAMINATHAN; BROOKS, 2019). A fórmula utilizada é apresentada na Equação 6.13. Na equação descrita $log_{\lambda ij}$ representa a taxa do *processo da distribuição negativa binomial* e ϵ_{ij} é o erro das variáveis dependentes γ_{1j}, γ_{2j} e γ_{3j}

$$(\log_{\lambda ij})\gamma_{ij} = \beta_0 + \beta_1(categoria_j) + \beta_2(comprimento_j) + \beta_3(hora\_postada_j) + \beta_4(dia\_semana_j) + \beta_5(analitico_j) + \beta_6(influencia_j) + \beta_7(subjetividade_j) + \beta_8(emocao_j)\epsilon_{ij}$$

Figura 6.13: Equação da regressão binomial não negativa gerada

Os resultados desta análise nos permitiram determinar quais são as características mais relevantes ao se gerar um modelo que represente um canal e os vídeos postados por ele. Ficou evidente também que diferentes fatores contribuem para que um vídeo gere likes e comentários. A princípio acreditávamos que somente o número de visualizações e inscritos do canal eram suficientes para determinar sua influência. Por fim, esta análise também mostrou que os usuários do Youtube tendem a apreciar vídeos com caráter argumentativo e informativo e com um tom mais confiante e polaridade neutra. Apesar deste resultado ser relevante para esta pesquisa ele contradiz outros trabalhos na literatura, como em (ALETI et al., 2019), (XU; ZHANG, 2018) e (PENNEBAKER et al., 2015), que defendem que os conteúdos mais apreciados e disseminados nas redes sociais, são os com uma alta valência (tanto positiva como negativa), além de possuírem uma narrativa envolvida. Podemos justificar este resultado argumentando que diferentemente de outras redes sociais, as pessoas no Youtube acessam esta rede com o objetivo de encontrar uma informação e/ou resolver um problema. Isto pode ser corroborado pela grande quantidade de tutorias e videoaulas disponíveis dos mais diversos temas.

Id	# **Vídeos e Proporção**	**Top 10 Palavras**	**Rótulo**	**Exemplo de Canais Associados:**
13	2937 8.49%	poder - galera - lutar - tempo - herói - novo - poderoso - forte - quadro - mundo	Cultura e Entretenimento	Ei Nerd, Bibi, Meteoro Brasil, Whindersson Nunes
26	1861 5.38%	bolar - bom - receitar - chocolate - massa - leite - amor - pouco - formar - minuto	Gastronomia	TPM por Ju Ferraz, Receitas da Cris, Receitas de Pai, Dani Noce
7	1589 4.60%	governar - bolsonaro - presidente - político - brasil - lula - público - estar - país - partir	Política, Economia e Notícias	Kim Kataguiri, TV Afiada, Mamaefalei, Nando Moura
0	1361 3.94%	pelar - maquiagem - base - pouco - produto - sombra - rostir - olhar - bom - tom	Beleza	Mariana Saad, Mari Maria, NiinaSecrets, Bianca Andrade
46	1323 3.83%	aparelhar - câmera - tela - foto - bom - bater - melhor - celular - samsung - iphone	Tecnologia	TudoCelular, Canaltech, Dudu Rocha, Be!Tech
40	1303 3.77%	legal - branco - ver - azul - vermelhar - marco - vez - piscina - bom - novo	Família	Brancoala, Flavia Calina, resendeevil, T3ddy
27	1199 3.47%	deus - bom - amor - foto - amigo - beijar - feliz - vidar - mundo - dia	Pessoas, Comportamento e Estilo de Vida	Taciele Alcolea, Central de fãs de Luisa Mell, Graciele Lacerda dia a dia, Evelyn Regly
17	1088 3.15%	partir - lado - cima - papel - pontar - colar - pronto - baixar - linha - pedaço	Decoração, Organização e DIY	Dany Martines, Paula Stephânia, Diycore com Karla Amadori, Manual do Mundo
42	941 2.72%	bom - gostoso - comido - água - pouco - carnar - café - prato - frango - queijar	Gastronomia	Tastemade Brasil, Dani Noce, Receitas de Pai, Sal de Flor
31	894 2.59%	parede - quartar - banheiro - portar - cozinhar - espaçar - sala - madeirar - mesa - cama	Decoração, Organização e DIY	Doma Arquitetura, Diycore com Karla Amadori, Organize sem Frescuras!, Maurício Arruda
34	837 2.42%	dinheiro - real - ano - mês - contar - banco - valor - investimento - taxar - pessoa	Economia, Empreendedorismo e Negócios	Me poupe!, O Primo Rico, Bruno Perini, Tiago Fonseca
44	812 2.35%	cachorro - animar - gatar - animal - bichar - espécie - peixe - gato - grande - maior	Pets e Animais	Richard Rasmussen, Estopinha & Alexandre rossi, Central de fãs de Luisa Mell, Você Sabia?
15	805 2.33%	viagem - lugar - hotel - legal - horar - avião - cidade - mundo - dólar - dia	Viagens, Aprendizados e Curiosidades	Estevam Pelo Mundo, Melhores Destinos, Prefiro Viajar, Viajo logo existo
14	772 2.23%	bolar - jogar - gol - time - desafiar - futebol - primeiro - copar - fred - bom	Esportes	Desimpedidos, Raquel Freestyle, Jogo Aberto, Denílson Show
3	763 2.21%	roupar - legal - lindo - loja - lindar - caixa - bonito - maravilhoso - presentar - bolsar	Moda e Estilo de Vida	Organize sem frescuras!, Taciele Alcolea, NiinaSecrets, Flavia Pavanelli

Tabela 6.4: Tópicos Rotulados

Tópico Id	**Categoria - Rótulo**	**Canais intitulados nestas categorias**	**Canais encontrados que abordam este tópico**
13	Cultura e Entretenimento	4	95
26	Gastronomia	7	43
7	Política, Economia e Notícias	5	31
0	Beleza	4	43
46	Tecnologia	5	15

Tabela 6.5: Principais tópicos criados *versus* categorias de origem dos canais

Argumento 1	**Relação**	**Argumento 2**
Jordan	ganhou_o_titulo_da	nba_91_92_93_96_97 98
Jordan	tem_tanto	título
que	que_da_para_ficar falando_tudo_em	um_vídeo
título	é	muita_coisa
esse_cara	tenha_sido_o_melhor_jogador	basquete

Tabela 6.6: Tuplas extraídas de um vídeo.

Título: O CINEMA BRASILEIRO EM NÚMEROS **Canal:** Meteoro Brasil			
#	**Argumento 1**	**Relação**	**Argumento 2**
1	o_total_de_filmes _brasileiros	lancados_comercialmente _de	1995_ate_2019_em _25_anos
2	vitoria_da_mangueira_no _carnaval_carioca_de_2016 _com_enredo _sobre_maria_bethania	mobilizou_um_publico_de	pouco_mais_de _18_mil_pessoas
3	o_cinema_nacional	atraiu_20_milhoes	mil_espectadores _em_2019
4	2003_carandiru	fez_publico_de	4_milhoes

Tabela 6.7: Exemplos de tuplas extraídas

Titulo do Vídeo: COMO VOU EMAGRECER 23KG EM 4 MESES! (E a relação entre dieta e investimentos!)		
Url: <https://www.youtube.com/watch?v=7U_2g1POsVI>		
Argumento 1	**Relação**	**Argumento 2**
dopamina	decida_que_é	um_habito
dopamina	comeca_não_é	um_habito
as_coisas	decida_que_é	um_habito
as_coisas	comeca_não_é	um_habito
um_convite_para_te	decida_que_é	um_habito
um_convite_para_te	comeca_não_é	um_habito

Tabela 6.8: Exemplo de tuplas extraídas e sem semântica

A biologia do Baby Yoda | Nerdologia **Link:** https://www.youtube.com/watch?v=0JkqXpefUt4		
Tempo do Vídeo:	**Transcrição Obtida:**	**Texto Correto:**
5:12	... nosso cérebro para falar **com vivem** grandes sociedades ...	... nosso cérebro para falar, **conviver em** grandes sociedades ...
5:20	... agora se o **iodo** é tão inteligente que ele precisa de décadas para ...	... agora se o **Yoda** é tão inteligente que ele precisa de décadas para ...
6:01	... que demanda menos batimentos por segundos a gente podia inclusive **vê** ainda mais...	.. que demanda menos batimentos por segundos a gente podia inclusive, **viver** ainda mais...
7:13	...equivalente **a continuado** ficaria na frente **jorge** rodrigues foi...	...equivalente **ao acolchoado,** ficaria na frente. **Daniel** Rodrigues foi...

Tabela 6.9: Alguns erros de speech recognition das transcrições

Experimento	**Tag Método**	**# Sentenças Sucesso**	**# Sentenças Falha**	**% Falha**
1	word2vec-70	1466	114	7.22%
2	bert-cosseno-70	1452	128	8.10%
3	word2vec-80	1394	186	11.77%
4	bert-cosseno-80	1333	247	15.63%
5	distancia-cosseno-70	1250	330	20.89%
6	distancia-cosseno-80	1043	537	33.99%
7	tsne_distortion	776	804	50.89%
8	embedding_distortion	659	921	58.29%

Tabela 6.10: Métodos de agrupamento de tuplas similares

Tag Método Avaliador1	**Tag Método Avaliador2**	**Total Grupo Av1**	**Total Grupo Av2**	**Grupos Iguais**	**%**
padrao_ouro _mcla	bert-cosseno-70	942	2145	428	45.44%
padrao_ouro _mcla	word2vec -80	942	2539	416	44.16%
padrao_ouro _mcla	bert-cosseno-80	942	2743	413	43.84%
padrao_ouro _mcla	word2vec -70	942	2039	387	41.08%
padrao_ouro _mcla	distancia-cosseno-70	942	3142	383	40.66%
padrao_ouro _mcla	distancia-cosseno-80	942	3476	338	35.88%

Tabela 6.11: Coeficiente de similaridade entre *cluster ensembles* e avaliadores humanos.

Limiar	**bert-cosseno-70**	**bert-cosseno-80**	**distancia-cosseno-70**	**distancia-cosseno-80**	**word2vec-70**	**word2vec-80**
0,75	48,73%	47,35%	42,14%	36,62%	44,27%	46,71%
0,80	47,03%	45,44%	40,98%	35,88%	42,78%	45,01%
0,85	45,97%	44,27%	40,98%	35,88%	41,61%	44,48%
0,90	45,65%	44,06%	40,76%	35,88%	41,30%	44,27%
0,95	45,44%	43,84%	40,66%	35,88%	41,08%	44,16%
1,00	45,44%	43,84%	40,66%	35,88%	41,08%	44,16%
Média	**46,38%**	**44,80%**	**41,03%**	**36,00%**	**42,02%**	**44,80%**

Tabela 6.12: Coeficiente de similaridade entre *cluster ensembles* e avaliadores humanos considerando limiares de similaridade a partir de 75%.

#	Url	Sentença	Patrocinado?	Tipo Transcrição
1	https://www.youtube.com/watch?v=BwTuQZPPtQ4&ab_channel=DuduRocha	de café incluindo **Nespresso** Orra, senti um **patrocínio** aí hein?	Não	Manual
2	https://www.youtube.com/watch?v=C4eHJ8ZJgG4	Esse nerdologia foi um **oferecimento** da **Kanui.com.br**	Sim	Manual
3	https://www.youtube.com/watch?v=HeWBW_MPiy0	palestras ao redor do **Brasil**, eu tenho os meus treinamentos e. cursos, eu tenho os meus livros, eu tenho **publicidade**, eu tenho o. adsense do **YouTube**, eu tenho participações minoritárias em negócios	Não	ASR
4	https://www.youtube.com/watch?v=_m3T7GJVJlI	esse lance é o **oferecimento** de **gilette** clube	Sim	ASR
5	https://www.youtube.com/watch?v=2DSsa7QXIi4	Esse vídeo é um **oferecimento** da **dell** e como você sabe, eu sou fã da marca faz tempo.	Sim	ASR

Tabela 6.13: Exemplo de sentenças e pseudo-sentenças obtidas

Url: https://www.youtube.com/watch?v=dJyJ77GkhBE			
Título: O Guia BÁSICO para começar a INVESTIR com POUCO DINHEIRO! (e do jeito CERTO! Sem pagar taxas)			
Sentença: Inclusive alguns deles nem pagam taxa, por exemplo na Rico que é parceira aqui no canal a taxa de corretagem pra fundo imobiliário é zero.			
#	**Argumento 1**	**Relação**	**Argumento 2**
1	Inclusive alguns deles	é zero	por exemplo na Rico que é parceira aqui
2	Inclusive alguns deles	é zero	por exemplo na Rico que é parceira aqui
3	Inclusive alguns deles	é zero	por exemplo na Rico que é parceira aqui
4	a Rico	é parceira	aqui
5	Inclusive alguns deles	é	zero, por exemplo en a Rico
6	a Rico	é	parceira

Tabela 6.14: Exemplos de Tuplas Extraídas - Transcrição Manual

Url: https://www.youtube.com/watch?v=2DSsa7QXIi4			
Título: O NOVO DELL XPS 13 é APAIXONANTE! COMPACTO e PODEROSO, vai SER o MEU NOVO NOTEBOOK!			
Sentença: Esse vídeo é um oferecimento da dell e como você sabe, eu sou fã da marca faz tempo.			
#	**Argumento 1**	**Relação**	**Argumento 2**
1	Esse vídeo	é um oferecimento	da dell
2	Esse vídeo	é um oferecimento	da dell e como você sabe, eu sou fã da ma...
3	Esse vídeo	é um oferecimento	da dell e como você sabe, eu sou fã da ma...
4	Esse vídeo	é um oferecimento	e como você sabe, eu sou fã da marca
5	Esse vídeo	é um oferecimento	e como você sabe, eu sou fã faz tempo

Tabela 6.15: Exemplos de Tuplas Extraídas - Transcrição ASR

Url: https://www.youtube.com/watch?v=ucGORMF-Aro			
Título: 5 Caminhos POSSÍVEIS pra quem PERDEU O EMPREGO na pandemia! (Não é receita mágica!)			
Sentença: nesse vídeo ai gente recado importante da náutica no caso sou eu esse vídeo tem o apoio da estácio que a nossa parceirasso em graduação e pós-graduação que tá ajudando todos os			
#	**Argumento 1**	**Relação**	**Argumento 2**
1	Esse vídeo	tem o apoio	de a estácio que a nossa parceirasso em graduação
2	Esse vídeo	tem	o apoio de a estácio que em graduação pós-graduação
3	eu	tem	o apoio de a estácio que a nossa parceirasso em graduação e pós-graduação que tá ajudando

Tabela 6.16: Exemplos de Tuplas Extraídas - Transcrição ASR

Variável	**Notação**	**Mínimo**	**Máximo**	**Média**	**Desvio Padrão**
Nº de likes	γ_{1j}	0	8.242.848	71.670,92	193.350,98
Nº de deslikes	γ_{2j}	0	591.233	1.900,13	9502,193
Nº de comentários	γ_{3j}	0	815.963	6.697,04	23.146,8
Pensamento Analítico	analitico_j	0	99	31,9724	19,61932
Foco externo	influencia_j	0	99	66,4849	20,12769
Valência	emocao_j	0	1	0,4581171	0,061071
Subjetividade	subjetividade_j	0	1	0,514227	0.067938

Tabela 6.17: Variáveis contínuas e estatística descritiva

Variável	**Notação**	**Categoria**	**N**	**%**
Hora da postagem do vídeo	hora_postada$_j$	1 - Horário Comercial	3033	27,10%
		0 - Horário não Comercial	8144	72,90%
Data da postagem do vídeo	dia_semana$_j$	1 - Dia da Semana	3854	34,50%
		0 - Final de Semana	7323	65,50%
Comprimento do vídeo	comprimento$_j$	Longo - (>20 min)	1415	12,70%
		Médio - (10-19 min)	4274	38,20%
		Curto - (<10 min)	5488	49,10%
Tamanho do vídeo	categoria$_j$	Viagem	755	6,18%
		Tecnologia e Negócios	914	8,20%
		Pais	874	7,80%
		Infantil	730	6,50%
		Lar	906	8,10%
		Gaming	1293	11,60%
		Comida	817	7,30%
		Saúde	1129	10,10%
		Moda	1142	10,20%
		Entreterimento	1265	11,30%
		Beleza	1352	12,10%

Tabela 6.18: Variáveis categóricas e estatística descritiva

Capítulo 7

Conclusão

Neste documento foi apresentado um método para a geração de grafos de conhecimento a partir de transcrições de áudio oriundos de vídeos do Youtube. Este método abrange desde a aquisição dos dados a partir da rede social mencionada, passando pelo pré-processamento dos dados brutos, extração de tuplas de conhecimento, geração dos modelos de *topic modeling*, até finalmente a geração de um modelo de representação em um grafo de conhecimento.

Ademais, além do próprio método desenvolvido, foram realizados trabalhos complementares visando testar a aplicabilidade da utilização de grafos de conhecimento a partir de vídeos do Youtube em diversos cenários.

Em uma primeira análise, os resultados obtidos nesta pesquisa, cumpriram os objetivos traçados na seção 1.2.2, além de validar as hipóteses levantadas na seção 1.3. Desta forma, este trabalho mostrou ser possível: (i) a extração e representação de conhecimentos de vídeos do Youtube em um grafo de conhecimento e (ii) o refinamento, ou seja, a melhoria dos conhecimentos extraídos no grafo.

Indo além, podemos citar como contribuições adicionais desta pesquisa, o desenvolvimento de um ferramenta de detecção automática de conteúdo patrocinado nos vídeos do Youtube (RODRIGUES; MUNARO; PARAISO, 2021), um novo método para inferência do número ideal tópicos na área de *topic modeling* (RODRIGUES; PARAISO, 2020), um framework para modelagem de características de viralidade em vídeos (MUNARO et al., 2020; MUNARO et al., 2021b), e por fim, um estudo sobre o impacto no engajamento da audiência de diferentes assuntos nos vídeos publicados (MUNARO et al., 2021a).

Também é importante mencionar que todos os códigos implementados, bem como as bases geradas estão disponíveis para acesso no GitHub e no site oficial do projeto.

Evidentemente, o método apresentado possui algumas limitações. Talvez a limitação que mereça mais atenção, é em relação a qualidade das transcrições de áudio obtidas

de forma automática (transcrições ASR). Conforme discutido no decorrer do trabalho estas transcrições podem conter erros semânticos, além da ausência de pontuações nos documentos. Estes dois desafios comprometem a qualidade das tuplas geradas. Algumas abordagens foram testadas para atacar estes problemas, como por exemplo utilizar um algoritmo de *boundary detection* para adicionar automaticamente as pontuações nos textos (LÓPEZ; PARDO, 2015). Infelizmente esta abordagem não respondeu da forma esperada. Desta forma, foi introduzido o conceito de *pseudo-sentenças* com o objetivo de permitir o processamento das tuplas de conhecimento, mesmo em documentos sem pontuações. Entretanto esta solução ainda é longe da ideal, sendo necessários novos testes e refinamentos no modelo gerado.

Por fim, é importante compreendermos estas limitações, como uma possibilidade de definirmos novos desafios em trabalhos futuros. Desta forma, estes novos trabalhos já foram iniciados, onde já esta sendo desenvolvido um novo método para o aprendizado automático de novos termos e conceitos com os conhecimentos armazenados no grafo, a partir dos comentários dos vídeos, titulo e descrição. Estes novos termos serão então agrupados baseado nos tópicos dos vídeos analisados. Finalmente, estes novos conceitos podem então ser utilizados na correção da transcrição de áudio gerada automaticamente.

Ainda como trabalho futuro, está sendo realizado um estudo em uma das maiores marcas de cosméticos do Brasil, com o objetivo de mapear a percepção e opinião dos consumidores e influenciadores digitais dos produtos produzidos por esta marca.

Capítulo 8

Anexos

Neste capítulo encontram-se todos os anexos citados anteriormente.

8.1 Tabela Entidades Nomeadas

Label	**Descrição**
PERSON	Nome de pessoas
NORP	Nacionalidades, religiões ou grupos políticos
FAC	Construções, aeroportos, pontes, etc.
ORG	Empresas, agências, instituições, etc.
GPE	Países, estados, cidades.
LOC	Localizações não-GPE
PRODUCT	Veiculos, objetos, comida, etc.
EVENT	Nome de guerras, batalhas, eventos esportivos, etc
WORK_OF_ART	Livros, músicas, obras de arte, etc
LAW	Documentos legais (ex. Constituição Brasileira)
LANGUAGE	Nomes de idiomas.
DATE	Datas ou períodos.
TIME	Tempos menores que um dia
PERCENT	Porcentagens (incluindo o simbolo '%')
MONEY	Valores monetários
QUANTITY	Quantidades, pesos, distâncias
ORDINAL	Números ordinais
CARDINAL	Números cardinais.

Tabela 8.1: Traduzido de: https://spacy.io/api/annotation

8.2 Stop Words Complementares

"aqui"	"gente"	"oi"	"então"	"caramba"	"aí"	"ufa"	"coisa"
"[Música]"	"né"	"tá"	"vídeo"	"casar"	"coisa"	"canal"	"tipo"
"mano"	"meio"	"ea"	"eo"	"se inscreva"	"ae"	"eis"	"caraca"
"joinha"	"like"	"só"	"siga"	"[Aplausos]"	"vezes"	"ai"	

Tabela 8.2: Lista de Stop Words Expandida.

8.3 Palavras-Chave Patrocínio

Palavras-chave:
Publicidade
Parceria
Parceria Paga
Conteúdo patrocinado
Apoio
Oferecimento
Patrocínio

Tabela 8.3: Lista de palavras-chave candidatas à patrocínio.

Referências Bibliográficas

ABDELALI, S. et al. Education data mining: Mining moocs videos using metadata based approach. In: IEEE. *2016 4th IEEE International Colloquium on Information Science and Technology (CiSt)*. [S.l.], 2016. p. 531–534.

ABREU, S. C. de; BONAMIGO, T. L.; VIEIRA, R. A review on relation extraction with an eye on portuguese. *Journal of the Brazilian Computer Society*, Springer, v. 19, n. 4, p. 553–571, 2013.

ADNAN, K.; AKBAR, R. Limitations of information extraction methods and techniques for heterogeneous unstructured big data. *International Journal of Engineering Business Management*, SAGE Publications Sage UK: London, England, v. 11, p. 1847979019890771, 2019.

AGARWAL, N.; LIU, H.; TANG, L.; YU, P. S. Identifying the influential bloggers in a community. In: *Proceedings of the 2008 international conference on web search and data mining*. [S.l.: s.n.], 2008. p. 207–218.

AGGRAWAL, N.; ARORA, A.; ANAND, A.; IRSHAD, M. View-count based modeling for youtube videos and weighted criteria–based ranking. In: *Advanced Mathematical Techniques in Engineering Sciences*. [S.l.]: CRC Press, 2018. p. 149–160.

ALETI, T.; PALLANT, J. I.; TUAN, A.; LAER, T. van. Tweeting with the stars: Automated text analysis of the effect of celebrity social media communications on consumer word of mouth. *Journal of Interactive Marketing*, Elsevier, v. 48, p. 17–32, 2019.

ALEXE, B.; HERNÁNDEZ, M. A.; HILDRUM, K. W.; KRISHNAMURTHY, R.; KOUTRIKA, G.; NAGARAJAN, M.; ROITMAN, H.; SHMUELI-SCHEUER, M.; STANOI, I. R.; VENKATRAMANI, C. et al. Surfacing time-critical insights from social media. In: *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. [S.l.: s.n.], 2012. p. 657–660.

ANGLES, R.; ARENAS, M.; BARCELÓ, P.; HOGAN, A.; REUTTER, J.; VRGOČ, D. Foundations of modern query languages for graph databases. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 50, n. 5, p. 1–40, 2017.

ARORA, A.; BANSAL, S.; KANDPAL, C.; ASWANI, R.; DWIVEDI, Y. Measuring social media influencer index-insights from facebook, twitter and instagram. *Journal of Retailing and Consumer Services*, Elsevier, v. 49, p. 86–101, 2019.

ARROYO-FERNÁNDEZ, I.; MÉNDEZ-CRUZ, C.-F.; SIERRA, G.; TORRES-MORENO, J.-M.; SIDOROV, G. Unsupervised sentence representations as word information series: Revisiting tf–idf. *Computer Speech & Language*, Elsevier, v. 56, p. 107–129, 2019.

BALABANIS, G.; CHATZOPOULOU, E. Under the influence of a blogger: The role of information-seeking goals and issue involvement. *Psychology & Marketing*, Wiley Online Library, v. 36, n. 4, p. 342–353, 2019.

BANERJEE, S.; CHUA, A. Y. Identifying the antecedents of posts' popularity on facebook fan pages. *Journal of Brand Management*, Springer, v. 26, n. 6, p. 621–633, 2019.

BARTKO, J. J. The intraclass correlation coefficient as a measure of reliability. *Psychological reports*, SAGE Publications Sage CA: Los Angeles, CA, v. 19, n. 1, p. 3–11, 1966.

BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. The semantic web. *Scientific american*, JSTOR, v. 284, n. 5, p. 34–43, 2001.

BERNSTEIN, A.; KAUFMANN, E.; KAISER, C. Querying the semantic web with ginseng: A guided input natural language search engine. In: CITESEER. *15th Workshop on Information Technologies and Systems, Las Vegas, NV*. [S.l.], 2005. p. 112–126.

BHUIYAN, H.; ARA, J.; BARDHAN, R.; ISLAM, M. R. Retrieving youtube video by sentiment analysis on user comment. In: IEEE. *2017 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*. [S.l.], 2017. p. 474–478.

BIRD, S.; KLEIN, E.; LOPER, E. *Natural language processing with Python: analyzing text with the natural language toolkit*. [S.l.]: "O'Reilly Media, Inc.", 2009.

BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent dirichlet allocation. *Journal of machine Learning research*, v. 3, n. Jan, p. 993–1022, 2003.

BOJANOWSKI, P.; GRAVE, E.; JOULIN, A.; MIKOLOV, T. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

BOLLACKER, K.; TUFTS, P.; PIERCE, T.; COOK, R. A platform for scalable, collaborative, structured information integration. In: *Intl. Workshop on Information Integration on the Web (IIWeb'07)*. [S.l.: s.n.], 2007. p. 22–27.

BONATTI, P. A.; DECKER, S.; POLLERES, A.; PRESUTTI, V. Knowledge graphs: new directions for knowledge representation on the semantic web (dagstuhl seminar 18371). In: SCHLOSS DAGSTUHL-LEIBNIZ-ZENTRUM FUER INFORMATIK. [S.l.], 2019.

BONDY, J. A.; MURTY, U. S. R. et al. *Graph theory with applications*. [S.l.]: Macmillan London, 1976. v. 290.

BOSER, B. E.; GUYON, I. M.; VAPNIK, V. N. A training algorithm for optimal margin classifiers. In: *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*. [S.l.: s.n.], 1992. p. 144–152.

CABRIO, E.; COJAN, J.; APROSIO, A. P.; MAGNINI, B.; LAVELLI, A.; GANDON, F. Qakis: an open domain qa system based on relational patterns. 2012.

CAMACHO-COLLADOS, J.; PILEHVAR, M. T. From word to sense embeddings: A survey on vector representations of meaning. *Journal of Artificial Intelligence Research*, v. 63, p. 743–788, 2018.

CHANG, J.; GERRISH, S.; WANG, C.; BOYD-GRABER, J. L.; BLEI, D. M. Reading tea leaves: How humans interpret topic models. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2009. p. 288–296.

CHANG, W.-L. Will sentiments in comments influence online video popularity? In: IEEE. *2018 IEEE International Conference on Big Data (Big Data)*. [S.l.], 2018. p. 3644–3646.

CIAMPAGLIA, G. L.; SHIRALKAR, P.; ROCHA, L. M.; BOLLEN, J.; MENCZER, F.; FLAMMINI, A. Computational fact checking from knowledge networks. *PloS one*, Public Library of Science, v. 10, n. 6, 2015.

CIMIANO, P.; HAASE, P.; HEIZMANN, J.; MANTEL, M.; STUDER, R. Towards portable natural language interfaces to knowledge bases–the case of the orakel system. *Data & Knowledge Engineering*, Elsevier, v. 65, n. 2, p. 325–354, 2008.

CLAUSET, A.; SHALIZI, C. R.; NEWMAN, M. E. Power-law distributions in empirical data. *SIAM review*, SIAM, v. 51, n. 4, p. 661–703, 2009.

CLEMENT, J. Hours of video uploaded to youtube every minute. *Statista. com*, 2019.

COLLOVINI, S.; MACHADO, G.; VIEIRA, R. Extracting and structuring open relations from portuguese text. In: SPRINGER. *International Conference on Computational Processing of the Portuguese Language*. [S.l.], 2016. p. 153–164.

CONRADO, M. d. S. *O efeito do uso de diferentes formas de extração de termos na compreensibilidade e representatividade dos termos em coleções textuais na língua portuguesa*. Tese (Doutorado) — Universidade de São Paulo, 2009.

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

DABBÈCHI, H.; HADDAR, N.; ABDALLAH, M. B.; HADDAR, K. A unified multidimensional data model from social networks for unstructured data analysis. In: IEEE. *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*. [S.l.], 2017. p. 415–422.

DAMLJANOVIC, D.; AGATONOVIC, M.; CUNNINGHAM, H. Freya: An interactive way of querying linked data using natural language. In: SPRINGER. *Extended Semantic Web Conference*. [S.l.], 2011. p. 125–138.

DARAGHMI, E. Y.; MING, Y. S. Using graph theory to re-verify the small world theory in an online social network word. In: *Proceedings of the 14th International Conference on Information Integration and Web-based Applications & Services*. [S.l.: s.n.], 2012. p. 407–410.

DAVIDSON, J.; LIEBALD, B.; LIU, J.; NANDY, P.; VLEET, T. V.; GARGI, U.; GUPTA, S.; HE, Y.; LAMBERT, M.; LIVINGSTON, B. et al. The youtube video recommendation system. In: *Proceedings of the fourth ACM conference on Recommender systems*. [S.l.: s.n.], 2010. p. 293–296.

DECKER, S.; MITRA, P.; MELNIK, S. Framework for the semantic web: an rdf tutorial. *IEEE Internet Computing*, IEEE, v. 4, n. 6, p. 68–73, 2000.

DEMŠAR, J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, v. 7, n. Jan, p. 1–30, 2006.

DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding. 10 2018.

DIAS, L. C. R. *O YouTube: potencialidades pedagógicas na aprendizagem da Língua Inglesa no 1. º ciclo do ensino básico*. Tese (Doutorado) — Instituto Politécnico do Porto. Escola Superior de Educação, 2013.

EULER, L. Solutio problematis ad geometriam situs pertinentis. *Commentarii academiae scientiarum Petropolitanae*, p. 128–140, 1741.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. *AI magazine*, v. 17, n. 3, p. 37–37, 1996.

FEOFILOFF, P.; KOHAYAKAWA, Y.; WAKABAYASHI, Y. Uma introdução sucinta à teoria dos grafos. 2011.

FRANCALANCI, C.; HUSSAIN, A. Influence-based twitter browsing with navigtweet. *Information Systems*, Elsevier, v. 64, p. 119–131, 2017.

FREITAS, A.; FARIA, F. F. de; O'RIAIN, S.; CURRY, E. Answering natural language queries over linked data graphs: a distributional semantics approach. In: *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. [S.l.: s.n.], 2013. p. 1107–1108.

FRIEDMAN, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, Taylor & Francis, v. 32, n. 200, p. 675–701, 1937.

FUNG, K. W.; BODENREIDER, O. Knowledge representation and ontologies. In: *Clinical research informatics*. [S.l.]: Springer, 2019. p. 313–339.

GANI, A.; SIDDIQA, A.; SHAMSHIRBAND, S.; HANUM, F. A survey on indexing techniques for big data: taxonomy and performance evaluation. *Knowledge and information systems*, Springer, v. 46, n. 2, p. 241–284, 2016.

GAUSBY, A. Attention spans. *Consumer Insights, Microsoft Canada*, 2015.

GERHARDS, C. Product placement on youtube: An explorative study on youtube creators' experiences with advertisers. *Convergence*, SAGE Publications Sage UK: London, England, v. 25, n. 3, p. 516–533, 2019.

GOEL, S.; WATTS, D. J.; GOLDSTEIN, D. G. The structure of online diffusion networks. In: *Proceedings of the 13th ACM conference on electronic commerce*. [S.l.: s.n.], 2012. p. 623–638.

GOOGLE. *What the world watched in a day*. 2019. Disponível em: <https://www.thinkwithgoogle.com/feature/youtube-video-data-watching-habits/>.

GRIBKOVSKAIA, I.; SR, Ø. H.; LAPORTE, G. The bridges of königsberg—a historical perspective. *Networks: An International Journal*, Wiley Online Library, v. 49, n. 3, p. 199–203, 2007.

GUIZZARDI, G. Ontological foundations for structural conceptual models. 2005.

HAGEN, L. Content analysis of e-petitions with topic modeling: How to train and evaluate lda models? *Information Processing & Management*, Elsevier, v. 54, n. 6, p. 1292–1307, 2018.

HARMAN, D. How effective is suffixing? *Journal of the american society for information science*, Wiley Online Library, v. 42, n. 1, p. 7–15, 1991.

HARTLEY, R. T.; BARNDEN, J. A. Semantic networks: visualizations of knowledge. *Trends in Cognitive Sciences*, Elsevier, v. 1, n. 5, p. 169–175, 1997.

HE, Q.; CHEN, B.; ARGAWAL, D. Building the linkedin knowledge graph. *LinkedIn*, 2016.

HEARST, M. A. Automatic acquisition of hyponyms from large text corpora. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 14th conference on Computational linguistics-Volume 2*. [S.l.], 1992. p. 539–545.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural computation*, MIT Press, v. 9, n. 8, p. 1735–1780, 1997.

HOFFART, J.; SUCHANEK, F. M.; BERBERICH, K.; LEWIS-KELHAM, E.; MELO, G. D.; WEIKUM, G. Yago2: exploring and querying world knowledge in time, space, context, and many languages. In: *Proceedings of the 20th international conference companion on World wide web*. [S.l.: s.n.], 2011. p. 229–232.

HOFMANN, T. Probabilistic latent semantic analysis. In: MORGAN KAUFMANN PUBLISHERS INC. *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. [S.l.], 1999. p. 289–296.

HOGAN, A.; BLOMQVIST, E.; COCHEZ, M.; D'AMATO, C.; MELO, G. D.; GUTIERREZ, C.; KIRRANE, S.; GAYO, J. E. L.; NAVIGLI, R.; NEUMAIER, S. et al. Knowledge graphs. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 54, n. 4, p. 1–37, 2021.

HOWARD, J.; RUDER, S. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

HUGHES, C.; SWAMINATHAN, V.; BROOKS, G. Driving brand engagement through online social influencers: An empirical investigation of sponsored blogging campaigns. *Journal of Marketing*, SAGE Publications Sage CA: Los Angeles, CA, v. 83, n. 5, p. 78–96, 2019.

HUTCHINS, J. The first public demonstration of machine translation: the georgetown-ibm system, 7th january 1954. 01 2004.

HUTTO, C. J.; GILBERT, E. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In: *Eighth international AAAI conference on weblogs and social media*. [S.l.: s.n.], 2014.

IOSUP, A.; HEGEMAN, T.; NGAI, W. L.; HELDENS, S.; PRAT-PÉREZ, A.; MANHARDTO, T.; CHAFIO, H.; CAPOTĂ, M.; SUNDARAM, N.; ANDERSON, M. et al. Ldbc graphalytics: A benchmark for large-scale graph analysis on parallel and distributed platforms. *Proceedings of the VLDB Endowment*, VLDB Endowment, v. 9, n. 13, p. 1317–1328, 2016.

JI, S.; PAN, S.; CAMBRIA, E.; MARTTINEN, P.; YU, P. S. A survey on knowledge graphs: Representation, acquisition and applications. *arXiv preprint arXiv:2002.00388*, 2020.

JURAFSKY, D. *Speech & language processing*. [S.l.]: Pearson Education India, 2000.

KARHAWI, I. et al. Influenciadores digitais: conceitos e práticas em discussão. *Communicare*, v. 17, p. 46–61, 2017.

KAUSHIK, L.; SANGWAN, A.; HANSEN, J. H. Automatic sentiment extraction from youtube videos. In: IEEE. *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. [S.l.], 2013. p. 239–244.

KHAN, G. F.; VONG, S. Virality over youtube: an empirical analysis. *Internet research*, Emerald Group Publishing Limited, 2014.

KLEINBERG, J. M. Hubs, authorities, and communities. *ACM computing surveys (CSUR)*, ACM New York, NY, USA, v. 31, n. 4es, p. 5–es, 1999.

KWON, J.; HAN, I.; KIM, B. Effects of source influence and peer referrals on information diffusion in twitter. *Industrial Management & Data Systems*, Emerald Publishing Limited, 2017.

LANDAUER, T. K.; MCNAMARA, D. S.; DENNIS, S.; KINTSCH, W. *Handbook of latent semantic analysis*. [S.l.]: Psychology Press, 2013.

LANGE, D.; BÖHM, C.; NAUMANN, F. Extracting structured information from wikipedia articles to populate infoboxes. In: *Proceedings of the 19th ACM international conference on Information and knowledge management*. [S.l.: s.n.], 2010. p. 1661–1664.

LASSILA, O.; SWICK, R. R. et al. Resource description framework (rdf) model and syntax specification. Citeseer, 1998.

LEE, D.; HOSANAGAR, K.; NAIR, H. S. Advertising content and consumer engagement on social media: Evidence from facebook. *Management Science*, INFORMS, v. 64, n. 11, p. 5105–5131, 2018.

LEHMAN, F. Semantic networks. *Computers and Mathematics with Applications*, v. 23, n. 2-5, p. 1–50, 1992.

LEHMANN, J.; ISELE, R.; JAKOB, M.; JENTZSCH, A.; KONTOKOSTAS, D.; MENDES, P. N.; HELLMANN, S.; MORSEY, M.; KLEEF, P. V.; AUER, S. et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, IOS Press, v. 6, n. 2, p. 167–195, 2015.

LI, X.; SHI, M.; WANG, X. S. Video mining: Measuring visual information using automatic methods. *International Journal of Research in Marketing*, Elsevier, v. 36, n. 2, p. 216–231, 2019.

LIN, Y.; LIU, Z.; SUN, M.; LIU, Y.; ZHU, X. Learning entity and relation embeddings for knowledge graph completion. In: *Twenty-ninth AAAI conference on artificial intelligence*. [S.l.: s.n.], 2015.

LIU, L.; TANG, L.; DONG, W.; YAO, S.; ZHOU, W. An overview of topic modeling and its current applications in bioinformatics. *SpringerPlus*, Springer, v. 5, n. 1, p. 1608, 2016.

LÓPEZ, R.; PARDO, T. A. Experiments on sentence boundary detection in user-generated web content. In: SPRINGER. *International Conference on Intelligent Text Processing and Computational Linguistics*. [S.l.], 2015. p. 227–237.

LOPEZ, V.; FERNÁNDEZ, M.; MOTTA, E.; STIELER, N. Poweraqua: Supporting users in querying and exploring the semantic web. *Semantic web*, IOS Press, v. 3, n. 3, p. 249–265, 2012.

LOPEZ, V.; UREN, V.; MOTTA, E.; PASIN, M. Aqualog: An ontology-driven question answering system for organizational semantic intranets. *Journal of Web Semantics*, Elsevier, v. 5, n. 2, p. 72–105, 2007.

LOVINS, J. B. Development of a stemming algorithm. *Mech. Translat. & Comp. Linguistics*, v. 11, n. 1-2, p. 22–31, 1968.

MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. *Introduction to information retrieval*. [S.l.]: Cambridge university press, 2008.

MARAJ, A.; MARTIN, M. V.; MAKREHCHI, M. A more effective sentence-wise text segmentation approach using bert. In: SPRINGER. *International Conference on Document Analysis and Recognition*. [S.l.], 2021. p. 236–250.

MARTINEZ-RODRIGUEZ, J. L.; HOGAN, A.; LOPEZ-AREVALO, I. Information extraction meets the semantic web: a survey. *Semantic Web*, IOS Press, n. Preprint, p. 1–81, 2018.

MATHEW, G.; SMITH, S. T.; PASSARELLI, J. Large scale open source video recommender tool using metadata surrogates. In: IEEE. *2018 IEEE International Conference on Big Data (Big Data)*. [S.l.], 2018. p. 1974–1977.

MIKOLOV, T.; CORRADO, G.; CHEN, K.; DEAN, J. Efficient estimation of word representations in vector space. In: . [S.l.: s.n.], 2013. p. 1–12.

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.

MILGRAM, S. The small world problem. *Psychology today*, New York, v. 2, n. 1, p. 60–67, 1967.

MINSKY, M. A framework for representing knowledge. 1974.

MISRA, H.; YVON, F.; JOSE, J. M.; CAPPE, O. Text segmentation via topic modeling: an analytical study. In: *Proceedings of the 18th ACM conference on Information and knowledge management*. [S.l.: s.n.], 2009. p. 1553–1556.

MIWA, M.; BANSAL, M. End-to-end relation extraction using lstms on sequences and tree structures. *arXiv preprint arXiv:1601.00770*, 2016.

MOODY, C. E. Mixing dirichlet topic models and word embeddings to make lda2vec. *arXiv preprint arXiv:1605.02019*, 2016.

MUNARO, A. C.; BARCELOS, R. H.; MAFFEZZOLLI, E. C. F.; RODRIGUES, J. P. S.; PARAISO, E. C. The drivers of video popularity on youtube: An empirical investigation. In: *Advances in Digital Marketing and eCommerce*. [S.l.]: Springer, 2020. p. 70–79.

MUNARO, A. C.; BARCELOS, R. H.; MAFFEZZOLLI, E. C. F.; RODRIGUES, J. P. S.; PARAISO, E. C. Mining meaning of videos on youtube: Unraveling latent content from digital influencers and their engagement. *Information Processing & Management*, Elsevier, 2021.

MUNARO, A. C.; BARCELOS, R. H.; MAFFEZZOLLI, E. C. F.; RODRIGUES, J. P. S.; PARAISO, E. C. To engage or not engage? the features of video content on youtube affecting digital consumer engagement. *Journal of Consumer Behaviour*, Wiley Online Library, 2021.

NEEDHAM, M.; HODLER, A. E. *Graph Algorithms: Practical Examples in Apache Spark and Neo4j*. [S.l.]: O'Reilly Media, 2019.

NEMENYI, P. Distribution-free multiple comparisons. In: INTERNATIONAL BIOMETRIC SOCIETY. *Biometrics*. [S.l.], 1962. v. 18, p. 263.

NICKEL, M.; MURPHY, K.; TRESP, V.; GABRILOVICH, E. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, IEEE, v. 104, n. 1, p. 11–33, 2015.

NOY, N.; GAO, Y.; JAIN, A.; NARAYANAN, A.; PATTERSON, A.; TAYLOR, J. Industry-scale knowledge graphs: Lessons and challenges. *Queue*, ACM New York, NY, USA, v. 17, n. 2, p. 48–75, 2019.

OLIVEIRA, L. S. de; GLAUBER, R.; CLARO, D. B. Dependentie: An open information extraction system on portuguese by a dependence analysis. *Encontro Nacional de Inteligência Artificial e Computacional*, 2017.

ORÚS, C.; BARLÉS, M. J.; BELANCHE, D.; CASALÓ, L.; FRAJ, E.; GURREA, R. The effects of learner-generated videos for youtube on learning outcomes and satisfaction. *Computers & Education*, Elsevier, v. 95, p. 254–269, 2016.

O'CALLAGHAN, D.; GREENE, D.; CARTHY, J.; CUNNINGHAM, P. An analysis of the coherence of descriptors in topic modeling. *Expert Systems with Applications*, Elsevier, v. 42, n. 13, p. 5645–5657, 2015.

PAGE, L.; BRIN, S.; MOTWANI, R.; WINOGRAD, T. *The PageRank citation ranking: Bringing order to the web.* [S.l.], 1999.

PAOLA, L. D.; RUVO, M. D.; PACI, P.; SANTONI, D.; GIULIANI, A. Protein contact networks: an emerging paradigm in chemistry. *Chemical reviews*, ACS Publications, v. 113, n. 3, p. 1598–1613, 2013.

PASTERNACK, J.; ROTH, D. Knowing what to believe (when you already know something). In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 23rd International Conference on Computational Linguistics*. [S.l.], 2010. p. 877–885.

PAULHEIM, H. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web*, IOS Press, v. 8, n. 3, p. 489–508, 2017.

PENNEBAKER, J. W.; CHUNG, C.; IRELAND, M.; GONZALES, A.; BOOTH, R. The development and psychometric properties of liwc2007. 2007. *URL: http://liwc. net/index. php [accessed 2015-09-14][WebCite Cache ID 6bX6QdwIO]*, 2015.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. [S.l.: s.n.], 2014. p. 1532–1543.

PETERS, M. E.; NEUMANN, M.; IYYER, M.; GARDNER, M.; CLARK, C.; LEE, K.; ZETTLEMOYER, L. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

PETERS, M. E.; NEUMANN, M.; IYYER, M.; GARDNER, M.; CLARK, C.; LEE, K.; ZETTLEMOYER, L. Deep contextualized word representations. In: *Proc. of NAACL*. [S.l.: s.n.], 2018.

PORTER, M. F. *An algorithm for suffix stripping, Readings in information retrieval*. [S.l.]: Morgan Kaufmann Publishers Inc., San Francisco, CA, 1997.

PUJARA, J.; MIAO, H.; GETOOR, L.; COHEN, W. Knowledge graph identification. In: SPRINGER. *International Semantic Web Conference*. [S.l.], 2013. p. 542–557.

QING, L.; MENGQIU, Y.; ZHAOPING, L.; JIANCHENG, L. Research on video automatic feature extraction technology based on deep neural network. In: IEEE. *2021 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*. [S.l.], 2021. p. 954–957.

RAMESH, I.; SIVAKUMAR, I.; RAMESH, K.; VENKATESH, V. P. P.; VETRISELVI, V. Categorization of youtube videos by video sampling and keyword processing. In: IEEE. *2020 International Conference on Communication and Signal Processing (ICCSP)*. [S.l.], 2020. p. 56–60.

RANGASWAMY, S.; GHOSH, S.; JHA, S.; RAMALINGAM, S. Metadata extraction and classification of youtube videos using sentiment analysis. In: IEEE. *2016 IEEE International Carnahan Conference on Security Technology (ICCST)*. [S.l.], 2016. p. 1–2.

REZENDE, S. O. *Sistemas inteligentes: fundamentos e aplicações*. [S.l.]: Editora Manole Ltda, 2003.

RODRIGUES, J. P.; PARAISO, E. From audio to information: Learning topics from audio transcripts. In: SBC. *Anais do VIII Symposium on Knowledge Discovery, Mining and Learning*. [S.l.], 2020. p. 121–128.

RODRIGUES, J. P. S.; MUNARO, A. C.; PARAISO, E. C. Identifying sponsored content in youtube using information extraction. In: IEEE. [S.l.], 2021.

ROSA, M. L.; FIANNACA, A.; RIZZO, R.; URSO, A. Probabilistic topic modeling for the analysis and classification of genomic sequences. *BMC bioinformatics*, BioMed Central, v. 16, n. 6, p. S2, 2015.

RUSSELL, S. J.; NORVIG, P. *Artificial intelligence: a modern approach*. [S.l.]: Malaysia; Pearson Education Limited,, 2016.

SABATE, F.; BERBEGAL-MIRABENT, J.; CAÑABATE, A.; LEBHERZ, P. R. Factors influencing popularity of branded content in facebook fan pages. *European management journal*, Elsevier, v. 32, n. 6, p. 1001–1011, 2014.

SALTON, G.; BUCKLEY, C. Term-weighting approaches in automatic text retrieval. *Information processing & management*, Elsevier, v. 24, n. 5, p. 513–523, 1988.

SCHWEMMER, C.; ZIEWIECKI, S. Social media sellout: The increasing role of product promotion on youtube. *Social Media+ Society*, SAGE Publications Sage UK: London, England, v. 4, n. 3, p. 2056305118786720, 2018.

SEBASTIANI, F. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, ACM New York, NY, USA, v. 34, n. 1, p. 1–47, 2002.

SEBASTIANI, F.; ESULI, A. Sentiwordnet: A publicly available lexical resource for opinion mining. In: *Proceedings of the 5th International Conference on Language Resources and Evaluation*. [S.l.: s.n.], 2006. p. 417–422.

SENA, C. F. L.; CLARO, D. B. Pragmatic information extraction in brazilian portuguese documents. In: SPRINGER. *International Conference on Computational Processing of the Portuguese Language*. [S.l.], 2018. p. 46–56.

SENEVIRATNE, C. N.; SAUER, C.; ROTH-BERGHOFER, T. Knowledge acquisition for the seasalt apprentice agent using twitter feeds. 2013.

SEUFERT, S.; ERNST, P.; BEDATHUR, S. J.; KONDREDDI, S. K.; BERBERICH, K.; WEIKUM, G. Instant espresso: Interactive analysis of relationships in knowledge graphs. In: *Proceedings of the 25th International Conference Companion on World Wide Web*. [S.l.: s.n.], 2016. p. 251–254.

SHAIKH, F.; PAWASKAR, D.; SIDDIQUI, A.; KHAN, U. Youtube data analysis using mapreduce on hadoop. In: IEEE. *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*. [S.l.], 2018. p. 2037–2041.

SHEKARPOUR, S.; NGOMO, A.-C. N.; AUER, S. Question answering on interlinked data. In: *Proceedings of the 22nd international conference on World Wide Web*. [S.l.: s.n.], 2013. p. 1145–1156.

SHRIVASTAVA, S. *Bring rich knowledge of people, places, things and local businesses to your apps. Bing Blogs*. 2017.

SIMÕES-PEREIRA, J. Grafos e redes: teoria e algoritmos básicos. *Rio de Janeiro: Interciência*, 2013.

SINGHAL, A. *Introducing the Knowledge Graph: things, not strings*. 2012. Disponível em: <https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html>.

SKIENA, S. S. *The algorithm design manual: Text*. [S.l.]: Springer Science & Business Media, 1998. v. 1.

SLEEMAN, J.; FININ, T. Type prediction for efficient coreference resolution in heterogeneous semantic graphs. In: IEEE. *2013 IEEE Seventh International Conference on Semantic Computing*. [S.l.], 2013. p. 78–85.

SLEEMAN, J.; FININ, T.; JOSHI, A. et al. Topic modeling for rdf graphs. In: *3rd International Workshop on Linked Data for Information Extraction, 14th International Semantic Web Conference*. [S.l.: s.n.], 2015. v. 1267, p. 48–62.

SOUZA, M. de; SOUZA, R. R. Modelagem de tópicos. *Múltiplos Olhares em Ciência da Informação*, v. 9, n. 2, 2019.

SOWA, J. F. Semantic networks. Citeseer, 1987.

STAPPEN, L.; BAIRD, A.; CAMBRIA, E.; SCHULLER, B. W. Sentiment analysis and topic recognition in video transcriptions. *IEEE Intelligent Systems*, IEEE, v. 36, n. 2, p. 88–95, 2021.

STEINBACH, M.; ERTÖZ, L.; KUMAR, V. The challenges of clustering high dimensional data. In: *New directions in statistical physics*. [S.l.]: Springer, 2004. p. 273–309.

STEVENS, K.; KEGELMEYER, P.; ANDRZEJEWSKI, D.; BUTTLER, D. Exploring topic coherence over many models and many topics. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. [S.l.], 2012. p. 952–961.

STREHL, A.; GHOSH, J. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, v. 3, n. Dec, p. 583–617, 2002.

SUCHANEK, F. M.; LAJUS, J.; BOSCHIN, A.; WEIKUM, G. Knowledge representation and rule mining in entity-centric knowledge bases. In: *Reasoning Web. Explainable Artificial Intelligence*. [S.l.]: Springer, 2019. p. 110–152.

SYAKUR, M.; KHOTIMAH, B.; ROCHMAN, E.; SATOTO, B. D. Integration k-means clustering method and elbow method for identification of the best customer profile cluster. In: IOP PUBLISHING. *IOP Conference Series: Materials Science and Engineering*. [S.l.], 2018. v. 336, n. 1, p. 012017.

SYED, S.; SPRUIT, M. Full-text or abstract? examining topic coherence scores using latent dirichlet allocation. In: IEEE. *2017 IEEE International conference on data science and advanced analytics (DSAA)*. [S.l.], 2017. p. 165–174.

THELWALL, M. Social media analytics for youtube comments: Potential and limitations. *International Journal of Social Research Methodology*, Taylor & Francis, v. 21, n. 3, p. 303–316, 2018.

TOMAN, M.; TESAR, R.; JEZEK, K. Influence of word normalization on text classification. *Proceedings of InSciT*, v. 4, p. 354–358, 2006.

TRAN, T.; WANG, H.; RUDOLPH, S.; CIMIANO, P. Top-k exploration of query candidates for efficient keyword search on graph-shaped (rdf) data. In: IEEE. *2009 IEEE 25th International Conference on Data Engineering*. [S.l.], 2009. p. 405–416.

TRSTENJAK, B.; MIKAC, S.; DONKO, D. Knn with tf-idf based framework for text categorization. *Procedia Engineering*, Elsevier, v. 69, p. 1356–1364, 2014.

TURIAN, J.; RATINOV, L.; BENGIO, Y. Word representations: a simple and general method for semi-supervised learning. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 48th annual meeting of the association for computational linguistics*. [S.l.], 2010. p. 384–394.

UNGER, C.; BÜHMANN, L.; LEHMANN, J.; NGOMO, A.-C. N.; GERBER, D.; CIMIANO, P. Template-based question answering over rdf data. In: *Proceedings of the 21st international conference on World Wide Web*. [S.l.: s.n.], 2012. p. 639–648.

UNGER, C.; CIMIANO, P. Pythia: Compositional meaning construction for ontology-based question answering on the semantic web. In: SPRINGER. *International conference on application of natural language to information systems*. [S.l.], 2011. p. 153–160.

UNGER, C.; FREITAS, A.; CIMIANO, P. An introduction to question answering over linked data. In: SPRINGER. *Reasoning Web International Summer School*. [S.l.], 2014. p. 100–140.

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A.; KAISER, L.; POLOSUKHIN, I. Attention is all you need. 06 2017.

VO, D.-T.; BAGHERI, E. Feature-enriched matrix factorization for relation extraction. *Information Processing & Management*, Elsevier, v. 56, n. 3, p. 424–444, 2019.

VRANDEČIĆ, D.; KRÖTZSCH, M. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, ACM New York, NY, USA, v. 57, n. 10, p. 78–85, 2014.

WAL, T. V. Folksonomy coinage and definition (2007). *Retrieved fromhttp: llvanderwal. netlfolksonomy. html. Accessed: December*, 2009.

WANDERLEY, G. M. P. *FolksDialogue: Um Método para o Aprendizado Automático de Folksonomias a partir de Diálogo Orientado à Tarefa em Português do Brasil*. Tese (Doutorado) — Pontifícia Universidade Católica do Paraná, 2015.

WANG, Q.; MAO, Z.; WANG, B.; GUO, L. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, v. 29, n. 12, p. 2724–2743, 2017.

WANG, Z.; ZHANG, J.; FENG, J.; CHEN, Z. Knowledge graph embedding by translating on hyperplanes. In: *Twenty-Eighth AAAI conference on artificial intelligence*. [S.l.: s.n.], 2014.

WATTS, D. J.; STROGATZ, S. H. Collective dynamics of 'small-world'networks. *nature*, Nature Publishing Group, v. 393, n. 6684, p. 440–442, 1998.

WEBSTER, J. J.; KIT, C. Tokenization as the initial phase in nlp. In: *COLING 1992 Volume 4: The 15th International Conference on Computational Linguistics*. [S.l.: s.n.], 1992.

WILBUR, W. J.; SIROTKIN, K. The automatic identification of stop words. *Journal of information science*, Sage Publications Sage CA: Thousand Oaks, CA, v. 18, n. 1, p. 45–55, 1992.

WITTEN, I. H.; FRANK, E.; HALL, M. A. Practical machine learning tools and techniques. *Morgan Kaufmann*, p. 578, 2005.

WÖLLMER, M.; WENINGER, F.; KNAUP, T.; SCHULLER, B.; SUN, C.; SAGAE, K.; MORENCY, L.-P. Youtube movie reviews: Sentiment analysis in an audio-visual context. *IEEE Intelligent Systems*, IEEE, v. 28, n. 3, p. 46–53, 2013.

WU, P.; ZHOU, Q.; LEI, Z.; QIU, W.; LI, X. Template oriented text summarization via knowledge graph. In: IEEE. *2018 International Conference on Audio, Language and Image Processing (ICALIP)*. [S.l.], 2018. p. 79–83.

XAVIER, C. C. et al. Learning non-verbal relations under open information extraction paradigm. Pontifícia Universidade Católica do Rio Grande do Sul, 2014.

XU, W. W.; ZHANG, C. Sentiment, richness, authority, and relevance model of information sharing during social crises—the case of# mh370 tweets. *Computers in Human Behavior*, Elsevier, v. 89, p. 199–206, 2018.

YAN, E.; DING, Y. Discovering author impact: A pagerank perspective. *Information processing & management*, Elsevier, v. 47, n. 1, p. 125–134, 2011.

YANG, Z.; DAI, Z.; YANG, Y.; CARBONELL, J.; SALAKHUTDINOV, R.; LE, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.

YOUTUBE. *YouTube in numbers*. 2020. Disponível em: <https://www.youtube.com/intl/en-GB/about/press/>.

ZAHN, C.; SCHAEFFELER, N.; GIEL, K. E.; WESSEL, D.; THIEL, A.; ZIPFEL, S.; HESSE, F. W. Video clips for youtube: Collaborative video creation as an educational concept for knowledge acquisition and attitude change related to obesity stigmatization. *Education and Information Technologies*, Springer, v. 19, n. 3, p. 603–621, 2014.

ZELENKO, D.; AONE, C.; RICHARDELLA, A. Kernel methods for relation extraction. *Journal of machine learning research*, v. 3, n. Feb, p. 1083–1106, 2003.

ZHANG, D. Y.; BADILLA, J.; TONG, H.; WANG, D. An end-to-end scalable copyright detection system for online video sharing platforms. In: IEEE. *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. [S.l.], 2018. p. 626–629.