UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

JONATA TEIXEIRA PASTRO

Detecção de *tweets* sobre Eventos de Trânsito usando Word Embedding

Monografia apresentada como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação

Orientador: Profa. Dra. Ana Lúcia Cetertich Bazzan

Porto Alegre
2018

Para Tamara, o meu trabalho
e a minha vida inteira.

AGRADECIMENTOS

À Prof. Dra. Ana Bazzan, por propiciar a realização deste trabalho.

À Dra. Clarissa Castellã Xavier, pela leitura e comentários.

Aos meus pais, Maria da Graça e Léo, sem vocês, nada seria possível.

À minha esposa, Tamara, por tudo.

À Débora e Idalina Coimbra, pelo apoio durante a execução desse trabalho.

RESUMO

O acesso à informações sobre incidentes de trânsito é importante tanto para os usuários quanto para órgãos reguladores de trânsito. Métodos tradicionais de detecção de incidentes requerem grande investimento em infraestrutura. Simultaneamente, usuários de mídias sociais como *Twitter* compartilham essas informações de forma autônoma. Contudo, a tarefa de identificar essas informações em meio ao grande volume de dados produzidos pelos usuários não é trivial. Considerando isso, o objetivo deste trabalho foi desenvolver um sistema de detecção de publicações na rede social *Twitter* sobre incidentes de trânsito. Foram selecionados *tweets* oriundos da cidade de Porto Alegre usando as informações de geolocalização disponível na interface de busca do *Twitter*. A aplicabilidade da técnica de *Word embedding* para classificação textual foi analisada, especificamente, as variantes *Doc2Vec* e *fastText*. Os resultados são apresentados por meio da comparação dos algoritmos utilizados, levando-se em conta questões como tempo de treinamento dos modelos e *F1-score* dos modelos.

Palavras-chave: Word Embedding. Classificação de Texto. Twitter. Trânsito.

Detection of tweets on Traffic Events using Word Embedding

ABSTRACT

Access to information on traffic incidents is important for both users and traffic regulators. Traditional methods of incident detection require large investment in infrastructure. At the same time, social media users as Twitter share this information autonomously. The results are presented by comparing the algorithms used, taking into account questions such as training time and accuracy of the models. The traffic events analyzed were selected from the city of Porto Alegre, geolocation system of *Twitter*. However, the task of identifying this information in the midst of the large volume of data produced by users is not trivial. Taking that into account, this work objective was developing a system of detection of publications in the social network Twitter about traffic incidents. The applicability of the Word embedding technique for text classification, specifically the Doc2Vec and fastText variants, was analyzed.

Keywords: Word Embedding. Text classification. Twitter. Traffic.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

1.1 Motivação

Alguns pesquisadores consideram que a aglomeração urbana está relacionada com o desenvolvimento econômico e social de uma sociedade (BLOOM; CANNING; FINK, 2008). Devido à isso, há uma tendência de urbanização em todo o mundo, com mais de metade de sua população morando em cidades em 2018, segundo (NATIONS, 2014). Ainda de acordo com esse relatório, 86% da população brasileira encontra-se em áreas urbanas, sendo São Paulo, por exemplo, a cidade mais populosa do país e a quarta maior do mundo, já Porto Alegre é a quinta região mais populosa do país.

Apesar das vantagens da urbanização, países em desenvolvimento passaram por esse processo de forma tardia e acelerada em comparação com os desenvolvidos, sem o estabelecimento de infraestrutura urbana equivalente (VASCONCELLOS, 2001), (GAKENHEIMER, 1999). Esta realidade, aliada com maior parte da população concentrada nas cidades, torna o acesso à mobilidade uma necessidade e seu planejamento e organização, um desafio.

A cidade de Porto Alegre é um exemplo dessa situação: entre 2007 e 2017 a frota de automóveis aumentou 30% totalizando 800 mil veículos, sendo quase 700 mil carros. A predileção por transporte individual combinada com a concentração do tráfego nas principais vias aumentam a frequência de congestionamentos (GAUCHAZH, 2017). Os problemas de mobilidade são evidenciados pela população, sendo apontado como maior problema de infraestrutura na cidade por 70% dos entrevistados, segundo pesquisa realizada pelo Sindicato Nacional de Arquitetura e Engenharia (Sinaenco) (G1, 2013).

Países em desenvolvimento possuem grandes desafios devido à falta de infraestrutura, planejamento e desconsideração pela acessibilidade oriundos do viés pelo transporte individual e a resistência em usar a experiência dos países desenvolvidos (GAKENHEIMER, 1999), (HUANG, 2003), (VASCONCELLOS, 2001). Zheng Dong Huang define o planejamento de transportes como a “formulação e avaliação de cenários por especialistas”. Sendo o “processo que inicia pela identificação de problemas de transporte e definição de objetivos e segue com os processos de previsão e avaliação”. Ainda segundo o autor, as atividades de planejamento e gerenciamento de transportes necessitam de grandes quantidades de dados (HUANG, 2003).

As fontes de dados tradicionais para planejamento de transportes incluem: pesqui-

sas à domicílio, local de trabalho ou em trânsito; monitoramento de tráfego, por sensores de velocidade, peso, imagem, dentre outros; mapeamento do uso da terra, infraestruturas e relevo; dados socioeconômicos (HUANG, 2003), (HAYDUK et al., 1997), (MILNE; WATLING, 2018).

Essas fontes apresentam desvantagens, como por exemplo, mapeamentos de uso de terra e dados socioeconômicos que apresentam uma imagem estática da situação; alto custo agregado as pesquisas, falta de respostas completas pelos entrevistados e o ambiente do trânsito dificulta a execução das pesquisas (HAYDUK et al., 1997), (AGRAWAL et al., 2015). O monitoramento de tráfego é espacialmente esparso, pois os sensores são instalados em apenas alguns pontos e de algumas vias (SAKAKI et al., 2012). Além disso, como aponta Huang, países em desenvolvimento possuem menos dados disponíveis de fontes tradicionais, o que em última instância acaba dificultando o planejamento de transportes (HUANG, 2003).

O uso de redes sociais aumenta a cada ano, por exemplo, no Brasil, em 2017, mais de 93% dos usuários de internet faziam parte de uma rede social, um aumento de 7% em relação ao período anterior de pesquisa, segundo estudo promovido pela agência *We are Social* (COELHO, 2018). Com o advento de dispositivos móveis, os dados de mídias sociais não são limitados a áreas pré-determinadas. Assim, os usuários podem compartilhar eventos que presenciam na momento e local que ocorrem. Essas características de localidade e instantaneidade tornam os dados coletados de mídias sociais muito ricos e importantes para o planejamento de transportes.

Enquanto sensores são excelentes em fornecer dados numéricos, como a extensão de um congestionamento, usuários em mídias sociais fornecem dados qualitativos diversos. Esses dados podem ser custosos ou impossíveis de identificar por sensores tradicionais. O grande volume de dados das mídias sociais permite que as evidências anedóticas dos usuários possuam também um caráter quantitativo.

Mídias sociais possuem escalas mais próximas da escala da população, seus usuários móveis estão distribuídos dinamicamente no espaço urbano e seu uso é frequente e contínuo, em contraponto dos métodos tradicionais de coleta de dados que se baseiam em amostras dispersas e/ou em determinados períodos.

Usuários de mídias sociais compartilham publicamente informações sobre suas atividades, inclusive sobre aquelas relacionadas a situações de trânsito, muitas vezes em tempo real. Se essas informações forem coletadas e apresentadas para usuários e operadores de trânsito, elas podem ser usadas para tomar decisões de rota mais eficientes ou

para apoiar sistemas de transporte inteligentes a controlar e planejar melhor o trânsito.

Dessa forma, como vantagens ao usar mídias sociais como monitores de tráfego, é possível elencar:

- Instantaneidade – mídias sociais permitem aos usuários noticiarem eventos no momento que acontecem. No caso de eventos de trânsito, elas permitem usuários reportarem incidentes ocorridos inclusive em áreas não cobertas por sensores de tráfego, com intuito de alerta outros usuários próximos de emergências;

- Representatividade – informações de trânsito coletadas por meio de sensores requerem análise automática ou humana para identificar incidentes de trânsito. Enquanto os usuários relatam diretamente os incidentes (CUI et al., 2014).

Conforme exposto, o planejamento de transportes necessita de grandes quantias de dados para análise. As mídias sociais trouxeram uma fonte ampla e, geralmente, aberta de informações. Existem plataformas que baseiam-se na colaboração do usuário para produção de conteúdo, o chamado "*crowdsourcing*". O exemplo mais emblemático é a Wikipédia, a enciclopédia livre, e na área de transportes, o aplicativo Waze, no qual os usuários compartilham as condições das vias e do tráfego. Mesmo em plataformas sem objetivo específico de colaboração, os usuários autonomamente compartilham informações de utilidade pública. Como exemplo, a plataforma Twitter possui a funcionalidade de marcação de mensagens com *hashtags* ou nomes de usuários que permitem identificar o objetivo da mensagem e facilitar a busca. Os usuários, por exemplo, em Porto Alegre, podem fazer postagens marcando o usuário @EPTC_POA, que é administrado pelo órgão de transporte público e trânsito da cidade, para reportar intercorrência no tráfego.

Mídias sociais, como o Twitter, possuem uma grande base de usuários e fornecem interfaces abertas de programação de aplicações para acessar os dados mais facilmente. Por isso, elas podem ser usadas como complementos para sensores de tráfego, auxiliando em seu monitoramento, e outros métodos tradicionais de identificação da condição do trânsito. Pois elas disponibilizam dados tanto qualitativos quanto quantitativamente ricos e permitem que usuários e operadores de trânsito possam responder a incidentes mais rapidamente. Além disso, as mídias sociais podem prover dados ausentes em outras fontes, como condições de trânsito para pedestres e ciclistas, ajudando a evitar o viés por veículos automotores, sobretudo individuais.

Para usar dados de mídias sociais para identificar incidentes de trânsito, primeiro é preciso selecionar quais postagens são relevantes. Sendo assim, podemos modelar o

problema como uma tarefa de classificação de texto. As características do texto de mídias sociais tornam essa tarefa desafiadora. Os textos são curtos, informais, com gírias e erros de grafia que dificultam filtrar o conteúdo por meio de busca por termos ou análise sintática. Sendo assim, é necessária uma técnica que permita identificar os textos pela similaridade semântica.

Uma das principais motivações desse trabalho é justamente manipular os dados disponíveis de forma aberta no Twitter e a caracterização de textos através do processamento de linguagem natural, uma vez que há poucos trabalhos feitos em português sobre o assunto, geralmente centrados em grupos de pesquisa como o NILC (USP-UFSCar) e o LXCenter de Portugal.

1.2 Objetivos

Neste trabalho a técnica de *word embedding* foi escolhida com o objetivo de classificar palavras e termos relacionados a eventos de trânsito no Twitter. Essa classe de algoritmos busca representar palavras no espaço vetorial de tal forma que palavras contidas em contextos semelhantes possuam representações vetoriais semelhantes. Mais especificamente foram escolhidas as variantes Doc2Vec e fastText que expandem o conceito de *word embedding* para textos, permitindo implementar classificadores.

O objetivo é analisar a aplicabilidade de algoritmos de *word embedding* para identificação de publicações informais curtas sobre eventos de trânsito. Para isso, foi desenvolvido um sistema capaz de coletar e classificar *tweets* como relacionados ou não a eventos de trânsito em Porto Alegre. A implementação permite escolher entre os algoritmos Doc2Vec e FastText para construção dos vetores de palavras/características e classificação.

1.3 Organização dos Capítulos

O presente trabalho se organiza da seguinte forma: inicialmente, o capítulo 2 apresenta o referencial teórico apresentando os principais conceitos. Em seguida, no capítulo 3 são apresentados os trabalhos relacionados em que são elencados as principais referências na área e discutidas de forma a contextualizar em que momento da pesquisa esse trabalho se insere e quais são as principais tendências até então. O capítulo 4 apresenta a

metodologia e a descrição pontual de cada etapa. O capítulo 5 apresenta os resultados e a avaliação dos mesmos. Por fim, as considerações finais no capítulo 6 apontam perspectivas do futuro da pesquisa nessa área, considerações finais e dificuldades encontradas.

2 REFERENCIAL TEÓRICO

Esse capítulo aborda as questões teóricas que são relevantes para o desenvolvimento desse trabalho, como os principais conceitos como Sistemas Inteligentes de Transporte , Twitter, Processamento de Linguagem Natural e *word embedding*.

2.1 Sistemas Inteligentes de Transporte

As principais questões trabalhadas no campo da Informática sobre Trânsito estão na área de Sistemas Inteligentes de Transporte que busca de forma ampla criar sistemas de trânsito mais inteligentes, menos subjetivos e mais focados nos dados e não só na estrutura.

É possível elencar várias razões que justificam a utilização de Sistemas Inteligentes de Transporte como a possibilidade de reduzir o tempo em trânsito em razão, principalmente, da melhoria de planejamento urbano; a redução de gasto de combustível o que auxilia no impacto ambiental; redução de fatalidades ao possibilitar uma análise mais objetiva e rápida de eventos que nem sempre a resposta humana conseguem suprir – ao mesmo tempo, que pode tratar de emergência de formas mais eficazes.

Além disso, os Sistemas Inteligentes de Transporte podem gerenciar congestionamentos de forma mais ampla do que alguns aplicativos como o Google Maps faz, com a otimização também de passageiros e de cargas; auxiliar o motorista em caso de mudanças climáticas repentinas (LEAL; SANTOS, 2015), (DARIDO, 2012).

Em grandes cidades, é possível ver o quanto o trânsito é um componente importante e até mesmo determinante da vida social e política. Em Porto Alegre, de acordo com matéria publicada pelo G1 em 2013 e se referindo a pesquisa encomendada pelo Sindicato Nacional de Arquitetura e Engenharia, a mobilidade é o grande problema da cidade. Em outro estudo mais recente, de 2017, aponta-se que houve um aumento de 30% dos automóveis em 10 anos na cidade sem que a infraestrutura municipal tenha acompanhado esse crescimento (G1, 2013).

Considerando esse contexto, a pesquisa utilizando Sistemas Inteligentes de Transporte é uma alternativa para sanar alguns desses problemas. Inicialmente, para definir esse conceito amplo podemos utilizar alguns autores como Williams que o define como sendo: “a aplicação de tecnologias de comunicação e computação em sistemas de transporte buscando o aumento da segurança e de sua eficiência” (WILLIAMS, 2008). Para

McQueen e McQueen, temos “a aplicação da tecnologia para melhorar a gerência e maximizar a utilização da infraestrutura de transportes, além da aplicação de tecnologias de computação e comunicação para operação e planejamento de transportes” (MCQUEEN; MCQUEEN, 1999). E ainda, Chowdhury classifica como uma:

> [...] variedade de ferramentas como conceitos de engenharia de tráfego, software, hardware e tecnologias de comunicação, que podem ser aplicadas de forma integrada ao sistema de transporte para melhorar sua eficiência e segurança. Podendo melhorar os diversos serviços que envolvem o gerenciamento do tráfego, principalmente através do potencial de reduzir o tempo de viagem, reduzir a frequência e gravidade dos acidentes, melhorar o fluxo, reduzir custos e melhorar a satisfação do cliente (CHOWDHURY; SADEK, 2003).

Sendo assim, os Sistemas Inteligentes de Transporte são compostos por soluções para controle e gerenciamento do tráfego por meio de serviços como definição dos horários e rotas de transporte público por meio de algoritmos que buscam facilitar e melhorar o transporte em cidades que apresentam esses problemas em seu cotidiano (DARIDO, 2012).

As possibilidades com os Sistemas Inteligentes de Transporte são extremamente variadas e ricas. Chowdhury e Sadek afirmam que o desenvolvimento de Sistemas Inteligentes de Transporte melhoram todas as formas de transportes de superfície e se atentam para a necessidade de que o planejamento da utilização dessa inteligência deve ocorrer tanto no nível nacional quanto em escalas menores. Os autores também apontam para a necessidade de profissionais com conhecimentos interdisciplinares, uma vez que para o manejamento do trânsito é necessário noções de fluxo de tráfego e aplicações de informática, por exemplo (CHOWDHURY; SADEK, 2003).

É possível perceber os diferentes usos dos Sistemas Inteligentes de Transporte voltados principalmente para a melhoria do trânsito e seus desdobramentos no cotidiano humano. Há também um crescente movimento de integração encontrado nas mídias sociais que permitem a maior conectividade e alcance de informações em tempo real.

Um termo essencial para o entendimento desse processo é o “*crowdsourcing*”, que pode ser definido como uma contribuição ou colaboração coletiva. Podendo resultar em uma abordagem que permita a substituição de sensores físicos ou de pessoas trabalhando diretamente para empresas de trânsito para se ter conteúdo desse tipo nas redes.

O “*crowdsourcing*” pode ser proativo, com as pessoas ligando ou se comunicando com e para as empresas de trânsito, avisando de intercorrências. Ou indireto, ocorrendo quando as pessoas decidem de forma autônoma compartilhar informações de trânsito, mas, não necessariamente para oferecer essas informações à órgãos, isso pode ser justificado também a partir do entendimento do porquê as pessoas compartilham conteúdos nas

mídias sociais e de que forma isso as enquadram no mundo em que vivem.

2.2 Mídias Sociais

Rede social pode ser entendida como uma produtora de redes uma vez que cria conexões de forma síncronas ou assíncronas, sendo sua principal característica a geração e circulação de informações o que possibilita novas formas de organização social integrando os interesses de diferentes coletivos. Já as mídias sociais permitem tanto a interação com outros membros quanto o compartilhamento de conteúdo. Um exemplo desse tipo são os blogs. Podemos tratar de forma bem específica a plataforma de micro-blog, o Twitter. Assim as "mídias sociais trabalham em ambientes relacionados" (BRAMBILLA et al., 2011) e ainda que são um

> [...] fenômeno complexo, que abarca o conjunto de novas tecnologias de comunicação mais participativas, mais rápidas e mais populares e as apropriações sociais que foram e que são geradas em torno dessas ferramentas. É um momento de hiperconexão em rede, onde estamos não apenas conectados, mas onde transcrevemos nossos grupos sociais e, através do suporte, geramos novas formas de circulação e difusão dessas informações. (BRAMBILLA et al., 2011)

Assim, é possível ainda fazer uma diferenciação entre diferentes tipos de mídia, havendo as tradicionais, emergentes e sociais. Sendo a primeira definida como portadora de informações de forma unilateral, como por exemplo, jornais, televisão, ou seja meios que não requerem a interação com o receptor. A segunda busca novas formas de interação entre empresas de comunicação e seus consumidores. E por fim, as mídias sociais correspondem a utilização das tecnologias de informação para estabelecer um canal de diálogo entre pessoas, empresas e afins.

Anita Whithing e David Williams utilizam uma abordagem baseada no uso e gratificação para compreender o porquê das pessoas utilizarem as mídias sociais, trabalhando igualmente com o conceito de consumidores. Dessa forma, apresentam dez gratificações que são levadas em conta pelos usuários, são elas: interação social, busca de informações, passatempo, entretenimento, relaxamento, utilidade comunicativa, utilidade de conveniência, expressão de opinião, compartilhamento de informações e vigilância / conhecimento sobre os outros. (WHITING; WILLIAMS, 2013)

2.2.1 Mídias Sociais como Fonte de Dados

As pesquisas com mídias sociais podem ser realizadas de diferentes formas. Uma delas as utiliza como ferramentas de pesquisa ou analisando as atividades e o seu conteúdo. A utilização dessa fonte ainda é considerada bastante recente na área e por isso, existem diferentes formas e metodologias de se trabalhar com elas, dependendo do objetivo do pesquisador.

Enquanto fonte de dados, é possível compreender o aumento do fluxo de informações e perceber que as redes sociais dizem respeito a pessoas e aos relacionamentos entre elas e não somente à tecnologia. Sua essência é a comunicação e os elementos que facilitam as interações (MARQUES; VIDIGAL, 2018). Enquanto fonte de informações, as mídias sociais nos oferecem uma variedade de dados que podem ser utilizados para diversos fins, como para pesquisas de mercado, utilizadas por empresas, como para pesquisas acadêmicas que visem o desenvolvimento de ferramentas e sistemas que dependam das informações dos usuários. Ainda nas palavras de Marques e Vidigal ao narrar sobre o contexto da pesquisa com cinco empresas brasileiras dos segmentos de educação, tecnologia, publicidade e consultoria, é possível perceber que:

> Baseando-se nos resultados da pesquisa e na literatura revisada, identificou-se que todas as empresas pesquisadas monitoram as redes sociais como uma fonte de informação mercadológica e estratégica, ou seja, detectou-se que estas são geradoras de dados diversos que podem auxiliar não somente o desenvolvimento de produtos e serviços, como também outros processos decisórios e setores da empresa. Porém, ressalta-se que é fundamental uma análise mais apurada, ou seja, uma verificação detalhada das redes para que as informações coletadas sejam realmente relevantes para os objetivos da empresa. (MARQUES; VIDIGAL, 2018)

Outra referência importante para pesquisa é apresentada por David Beer em seu trabalho "*Using Social Media Data Aggregators to Do Social Research*", em que ele apresenta o uso de agregadores – SMDA, *Social Media Data Aggregators* - que capturam e facilitam a análise dos dados acumulados nesses servidores. E o autor aborda também uma forma de utilizar esse tipo de ferramenta para a pesquisa científica. Beer aponta que ainda é preciso ter algumas precauções ao tratar das mídias sociais por conta da "tecnologização", ou seja, do uso da tecnologia nas pesquisas e até mesmo dela se tornar as fontes e objetos de investigação. Além disso, devemos compreender como os dados estão sendo ordenados para nós através de uma análise *a priori* analítica e o poder dos algoritmos para lidar, principalmente, com base de dados incompreensíveis e ainda buscar padrões em seu interior. (BEER, 2012)

2.2.2 Twitter

O *Twitter* é uma mídia social, usualmente denominado de *microblogging*, apesar de haverem divergências sobre essa denominação. Ela é utilizada por ter algumas ferramentas semelhantes aos blogs, mas, se difere bastante deles por sua estrutura de publicação. Ele foi fundado em 2006 e sua principal função é enviar e receber mensagens de atualização, os *tweets*, dos contatos que o usuário ativamente deseja seguir. Está disponível nas plataformas para celulares, tanto em sistemas operacionais de dispositivos móveis, como o sistema Android e o iOS, quanto em seu *website* e em um sistema de mensagens de textos do celular.

Até o ano passado, 2017, eram permitidas postagens com 140 caracteres, número que foi dobrado em uma das últimas atualizações do serviço o que também gerou algumas críticas dos usuários pelo novo formato adotado. Um de seus diferenciais é permitir a troca dessas mensagens em tempo real o que permite seu uso para criação de estatísticas em redes sociais e também sua utilização em pesquisas, uma vez que os usuários podem criar *tweets* sobre qualquer assunto de seus interesses.

Essa mídia social é popular como fonte de dados devido sua política de disponibilização de dados. A possibilidade do uso gratuito de sua API (*Application Program Interface*), permite que diferentes programas acessem sua base de dados mais facilmente, assim como é feito nesse trabalho.

Dessa forma, os trabalhos com mídias sociais ainda apresentam várias metodologias de abordagem e de pesquisa, ainda que sejam em sua maioria vistos a partir de seu grande potencial, uma vez que podem ser utilizadas por diferentes vertentes do conhecimento, como para o marketing, como para a computação, por exemplo. A partir disso, apresentamos o *Twitter* como uma mídia social amplamente utilizada e que constitui algumas vantagens quando pensada como fonte de dados para pesquisa.

2.3 Processamento de Linguagem Natural

Mídias sociais são fontes de dados significativas e baratas com aplicações comerciais e científicas. Seu uso depende da análise do texto escrito livre e também é possível utilizar recursos como a geolocalização, por exemplo. A complexidade da língua humana torna essa tarefa complicada, uma vez que a interação humana com a máquina não ocorre de forma direta. Não havendo o diálogo entre a linguagem computacional e o pensamento

humano, não há uma solução definitiva. Por isso, o campo de pesquisa em processamento de linguagem natural é extremamente ativo e dinâmico.

Esse campo possui duas abordagens principais: métodos baseados em conhecimento e métodos baseados em dados ou estatísticos. Inicialmente, os esforços de pesquisa estavam focados na primeira abordagem e visavam formalizar a linguagem humana por meio de regras definidas e lógica. De acordo com Sandra Martins, as "interfaces em linguagem natural surgem para proporcionar às pessoas uma forma de comunicação mais natural e simples com as máquinas" (MARTINS et al., 2004).

O foco em métodos baseados em conhecimento se deve à influência de teorias linguísticas como a gramática gerativa desenvolvida por Noam Chomsky e outros pesquisadores a partir de 1957. Sendo essa uma forma de estudar gramática que privilegia um aspecto específico da linguagem, ocupando-se preferencialmente da sintaxe das línguas objetivando uma entidade teórica chamada Gramática Universal. (VITRAL, 1996) Essa teoria busca descrever uma língua como um sistema de regras que gere as combinações de palavras que formem frases gramaticalmente válidas. Sob o princípio que a linguagem humana é derivada de uma gramática universal, ou seja, certas regras estruturais de linguagem são inatas aos humanos e independentes de sua experiência social.

Esses estudos tratavam de linguística computacional, a disciplina cientifica que estuda processos linguísticos por meio de uma perspectiva computacional com tópicos como compreensão da linguagem por psicolinguística computacional, produção de linguagem e aquisição de linguagem. Os métodos baseados em conhecimento possuem vantagens por aproveitar os estudos em linguagens formais, também necessários para o desenvolvimento de linguagens de programação. Além disso, esses sistemas concentravam a complexidade na definição das regras em vez de em tempo de execução, isso resultava em um baixo custo computacional.

Os métodos baseados em dados estatísticos se popularizaram a partir dos anos 1990 e são baseadas em análises linguísticas que almejam o mapeamento de diferentes *corpora* para formulações de traduções, aprendizado supervisionado e avaliação empírica do conhecimento dos sistemas (BROWN et al., 1993). O que representa uma mudança para entusiasmo e otimismo sobre a inteligência artificial (DAVIES, 2006). As bases de dados passaram a ser anotadas, tendo como exemplo, The Penn Treebank (MARCUS; MARCINKIEWICZ; SANTORINI, 1993). O luso-brasileiro, Floresta Sintá(c)tica (AFONSO et al., 2002).

Além disso, houve um aumento e difusão do poder computacional com a populari-

zação dos computadores pessoais, ao mesmo tempo em que houve uma perda de interesse na criação de sistemas com aplicações semânticas de alto nível, assim, aumentou-se o foco, reduzindo-se o escopo das categorizações de texto, identificação de função semântica, identificação de entidade nomeada e análise sintática.

O início do século XXI veio acompanhado da modelagem estatística e de algoritmos de máquina mais sofisticados. O aumento do conteúdo da web fez com as pesquisas utilizassem recursos de *big data* e o foco das aplicações semânticas voltaram a ser de alto nível, havendo também a mudança do foco para aprendizado não supervisionado.

Para perceber como ocorre o aprendizado de máquinas é necessário inicialmente, atentar para os três tipos principais de análise da linguagem que podem ser utilizadas em Língua Portuguesa, são elas: Análise Morfológica; Análise de Sintaxe e Análise Semântica.

2.3.1 Análise Morfológica

Inserida no campo gramatical, essa análise se concentra nas estruturas, formações e classificações da palavras isoladamente, inseridas ou não dentro de um contexto textual, buscando compreendê-las dentro de sua classe gramatical que pode ser: substantivo, adjetivo, artigo, pronome, numeral, verbo, advérbio, preposição, conjunção ou interjeição (CEGALLA, 2009), (PERINI et al., 1998).

Esse tipo de análise é importante, uma vez que permite a caracterização das palavras de acordo com sua classificação e possibilita que sejam permitidos padrões idiomáticos, por exemplo, estruturas frasais que sigam a ordem, no caso da Língua Portuguesa, de: sujeito + verbo + objeto, sendo que o sujeito da frase deve ser um substantivo que pode ser qualificado com um adjetivo, assim como o objeto, já o verbo deve indicar uma ação. Sendo assim, é possível criar padrões que facilitam o processo de aprendizado do idioma (BRASIL, 1998).

2.3.2 Análise de Sintaxe

A Sintaxe constitui junto com outros elementos como a semântica as formas de linguagem e que em linhas gerais busca compreender as palavras e as estruturas que compõem as frases, de acordo com Maria Helena P. Santos é possível defini-la da seguinte

forma:

> Cunha e Cintra (1986, p. 119-120) define 'sintaxe' como a "[...] parte da gramática que descreve as regras segundo as quais as palavras se combinam para formar frases [...]", reconhecendo, porém, no âmbito de uma pequena observação expendida no seio de um dos capítulos dedicados a essa área de descrição da língua, que o seu estudo implica "[...] o conhecimento de alguns conceitos nem sempre fáceis de definir [...]", o que decorre, na sua opinião, não só da [...] própria natureza do assunto, mas também das diferenças dos métodos e técnicas de análise adoptados pela Linguística Clássica e pelas principais correntes da Linguística Contemporânea. (SANTOS, 2014)

Essa é uma parte essencial para o entendimento desse trabalho, assim como a semântica, uma vez que são apresentados os conceitos de linguagens e para o processo de linguagens, sendo uma área que naturalmente interdisciplinar por tratar usufruir tanto de questões gramaticais e das letras, quanto das questões computacionais ao trazer essas questões para o meio digital e fazer com que o computador compreenda e apreenda essas técnicas.

O processo de interpretar um texto é bastante complexo e não existe uma definição formal de como as pessoas fazem isso e para poder reduzir a complexidade da língua que é naturalmente ambígua, imprecisa, tem vários processos sobre o texto que facilitam isso. Alguns processos devem ser levados em conta, por exemplo a lematização e redução ao radical, em inglês "*stemming*". O primeiro busca agrupar o padrão de diversos variantes de uma mesma expressão para facilitar a apresentação desse, um exemplo disso, é a apresentação de termos em dicionários. De acordo com Gallison: "Nos dicionários práticos, a lematização consiste em encontrar um item, isto é, uma forma gráfica representativa de todas as formas que uma unidade de significação lexicográfica (tradicionalmente palavra ou palavras compostas) pode tomar" (COSTE; GALLISON, 1983).

É importante ressaltar que essa é uma separação muitas vezes léxica e também é compreendida como um modo "convencional" de agrupamento de formas. Ou seja, o termo se refere aos radicais das palavras, a parte que transmite seu significado, para isso é necessário retirar os outros elementos das palavras como sufixos e prefixos que não fazem parte da sua matriz. O que difere lematização de *stemming* para J. L. De Lucca e Maria das Graças Volpe Nunes é:

> Lematização difere fundamentalmente de *stemming*. Enquanto lematização existe puramente no contexto lexicográfico, *stemming* não. Lematização é o singular, adjetivos e substativos e infinitivo (verbos), apenas no contexto da lexicologia. Stemming é a retirada de sufixos do radical, enquanto stem é o radical. Assim as estruturas são distintas, embora eventualmente possam ser graficamente semelhantes (LUCCA; NUNES, 2002).

Outro conceito importante é o de segmentação textual. Irma Kappel apresenta algumas contribuições sobre definições sobre essa forma de se analisar textos a partir de

sua divisão em segmentos. Para ela, existem algumas propriedades definidoras para isso, como por exemplo a centração, relevância, pontualização, organicidade para estabelecer os segmentos e os padrões em que eles se organizam dentro das estruturas discursivas (KAPPEL, 1998).

Há alguns padrões em que devem ser feitos para que os segmentos apresentem sentido. Outro trabalho feito sobre esse tema é do Núcleo Interinstitucional de Linguística Computacional (NILC) que fez uma revisão bibliográfica sobre segmentação textual automática, visando o desenvolvimento de um analisado discursivo para o português brasileiro - antes da reforma ortográfica de 2009, que unificou todas as variações do idioma.

A principal ideia desse conceito é "quebrar" o texto em elementos, partindo do mesmo princípio utilizado para a alfabetização de crianças, para perceber que há a necessidade de haver espaço entre as palavras, pontuação, parágrafos e assim por diante, quando trazemos isso para o âmbito computacional, é necessário criar "tokens", isso é, marcadores que façam o programa identificar os limites do texto e isso pode ser pensado de uma forma bem ampla, tanto limite físico de um texto quanto limite de assunto, ou seja a compreensão de várias temáticas dentro de um mesmo texto. Sendo assim, há várias possibilidades de trabalho com esse tipo de assunto.

Por fim, é necessário pensar na análise sintática das palavras, uma vez que é essencial compreender as classes gramaticais para poder classifica-las e também pensar na função das palavras e nas estruturas das frases.

2.3.3 Análise Semântica

A Semântica, por sua vez, estuda o significado das palavras, incidindo diretamente sobre a relação que se estabelece entre este e o significante. É focado especificamente nas questões usadas por seres humanos para se expressarem através da linguagem e de certa forma, se contrapõe à análise anterior, uma vez que uma foca na estrutura *per se* enquanto essa nas relações significativas forjadas.

A Semântica léxica se ocupa em tentar descobrir o sentido das palavras, subunidades da palavras, como sufixos, prefixos e afins, palavras compostas, como "pé-de-moleque" ou nomes de cidades como "Rio de Janeiro", "São Paulo" e expressões idiomáticas, principalmente pensando nas gírias que dependem de várias palavras para ser possível compreender o real significado. Assim, ela estuda a classificação e decomposição dos lexemas, ou seja, as unidades léxicas.

Os relacionamentos léxicos podem ser classificados como hiponímia e hiperonímia sendo o primeiro a relação entre um termo genérico e uma instância específica dele, por exemplo, a palavra “animal” que possui uma relação de hiponímia com “cachorro”, “gato”, “humano”, já a segunda é a relação simétrica entre o termo específico e todos os seus genéricos relacionados, por exemplo, um “gato” é um “felino”, é um “animal” é um “ser vivo”.

2.3.4 *N*-gramas

É importante pensar na ideia de *n*-gramas de um texto, ou seja, nas unidades mínimas que o computador pode interpretar e agrupar palavras ou expressões. Um *n*grama pode ser considerando tanto como palavra quanto como letra, havendo, dessa forma, dois escopos. Por exemplo, se pensarmos na frase “O trânsito está bom”, sendo N=2, podemos identificar os *n*-gramas como:

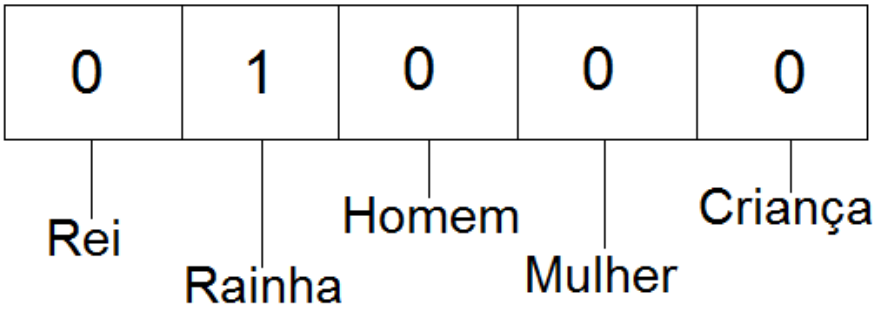
1. O trânsito;
2. trânsito está;
3. está bom;

Além disso, é possível prever a probabilidade de recorrência de um determinado grupo de palavras com base no *n*-grama (NASCIMENTO et al., 2012). De acordo com Jurafsky e Martin, ele funciona em um paradigma de treinamento e teste, utilizado nesse trabalho, que permite ensinar ensinar ao classificador as sequências de palavras associadas com determinados assuntos e classificação (JURAFSKY; MARTIN, 2014).

2.4 Representações de Palavras

Palavras são as unidades semânticas básicas em processamento de linguagem natural. A representação mais direta de uma palavra é uma cadeia de caracteres ou *string*. Enquanto *strings* são úteis para tarefas como análise morfológica, essa representação não auxilia a expressar o sentido das palavras. A similaridade na grafia não indica similaridade semântica, um exemplo são as palavras: cessão, seção e sessão. Além disso, palavras sinônimas podem ser grafadas de formas muito distintas, por exemplo, seção e

Figura 2.1: Codificação 1 de N de Rainha - Representação Local



Fonte: Os Autores

departamento. Manipulações sobre *strings* são menos eficientes do que sobre números, pois computadores são mais otimizados para trabalhar com os últimos.

2.4.1 Vetores de Palavras

O princípio de vetores de palavras é representar os elementos de um vocabulário por meio de vetores numéricos. As ciências cognitivas apontam duas abordagens para representar palavras no espaço vetorial: representação local e distribuída. (WILSON; KEIL, 2001)

Na representação local, cada elemento é atribuído a uma dimensão. Sendo assim, uma palavra é codificada como um vetor no qual o valor da sua dimensão correspondente é igual a um e os valores das outras dimensões são zero. No contexto de redes neurais, pode-se pensar que cada elemento do vocabulário é mapeado para uma unidade representacional. Por causa desse formato, ela é mais conhecida na área de processamento de linguagem natural pelos termos codificação 1-de-N ou *one-hot*. Na figura 2.1 pode-se ver uma representação 1-de-n da palavra rainha em um vocabulário de 5 palavras. Essa representação é extremamente esparsa e não possui propriedades semânticas.

Na representação distribuída, em vez de existir um mapeamento direto de uma dimensão para uma palavra, a representação desta é distribuída pelo vetor e o valor de cada dimensão é um número real. Cada elemento do vetor nessa abordagem representa uma propriedade da palavra. As representações são densas, contendo dezenas ou centenas de dimensões, em vez de milhares ou milhões necessárias em representações 1-de-N.

Tabela 2.1: Representações Distribuídas

	Princesa	Rei	Crianças	Rainha	Homem
Nobreza	0,97	0,98	0,01	0,99	0,02
Feminino	0,95	0,02	0,07	0,96	0,06
Idade	0,1	0,85	0,1	0,84	0,5
Plural	0,03	0,05	0,95	0,04	0,06
...	...	...	...	...	...

Fonte: Os Autores

2.4.2 *Word Embbeding*

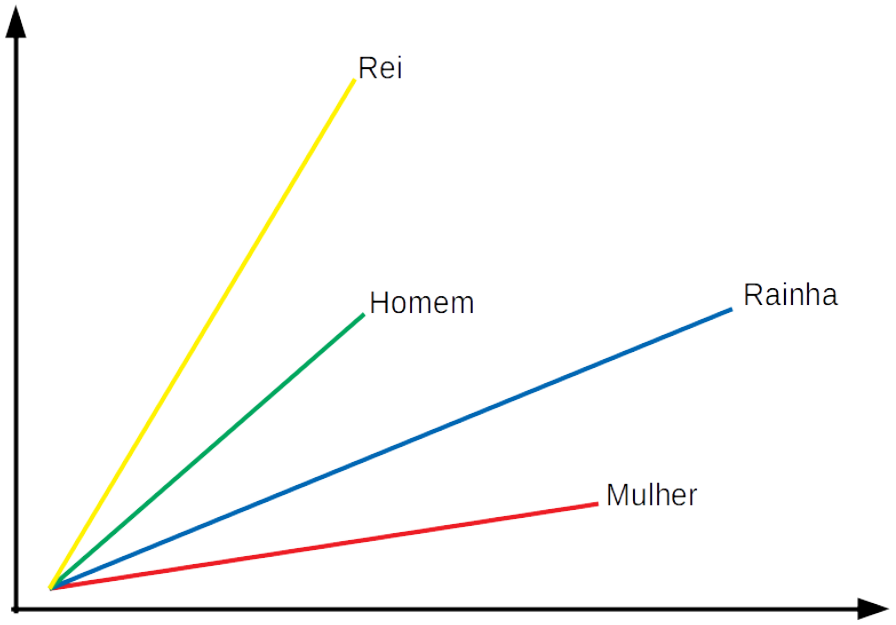
O termo *word embedding* refere-se a um conjunto de técnicas para obter representações distribuídas de palavras e expressões. *word embedding* é baseado no princípio matemático de imersão, ou seja, do mapeamento do espaço de maior dimensionalidade, por exemplo uma representação 1-de-N, para um menor. As técnicas buscam mapear palavras com significados semelhantes para representações semelhantes. Em "*Neural Probabilistic Language Models*", Bengio *et al* introduzem a abordagem mais comum para geração de *word embeddings*: aliar redes neurais e semântica distributiva. Esse conceito é derivado da hipótese distributiva de Zellig Harris, teoria linguística que propõe que palavras aparecendo em contextos similares possuem significados similares. (HARRIS, 1954) O aprendizado de *word embeddings* usando redes neurais pode ser pensada como uma tarefa autossupervisionada, onde as representações de palavras são aprendidas de amostras de uso das palavras na língua. Uma técnica usual é usar um grande corpus textual como, por exemplo, a Wikipédia.

Apesar de não haver uma rotulação explícita das dimensões dos vetores de palavra aprendidos, eles são capazes de capturar relacionamentos sintáticos e semânticos entre palavras. Pode-se imaginar uma rotulação hipotética dos vetores como visto na figura 2.4.2.

Essas representações podem ser usadas como vetores de características para algoritmos de classificação de texto. *Word embeddings* são capazes de expressar relações sintáticas e semânticas por meio de operações de aritmética vetorial (MIKOLOV; YIH; ZWEIG, 2013). Por exemplo, a relação entre masculino e feminino por ser capturada pela analogia: "homem está para rei assim como mulher está para rainha". Em aritmética vetorial essa relação é expressa pela equação 2.1.

$$V_{rei} - V_{homem} + V_{mulher} \approx V_{rainha} \tag{2.1}$$

Figura 2.2: Vetores de Palavras



Fonte: Os Autores

Onde V_p é a representação vetorial da palavra p. A relação de similaridade pode ser resolvida por distância de cossenos entre os vetores de palavras.

Esses vetores de palavra podem ser representados em projeção bidimensional conforme figura 2.2. Enquanto a composição vetorial expressa na equação 2.1 pode ser visualizada na figura 2.3.

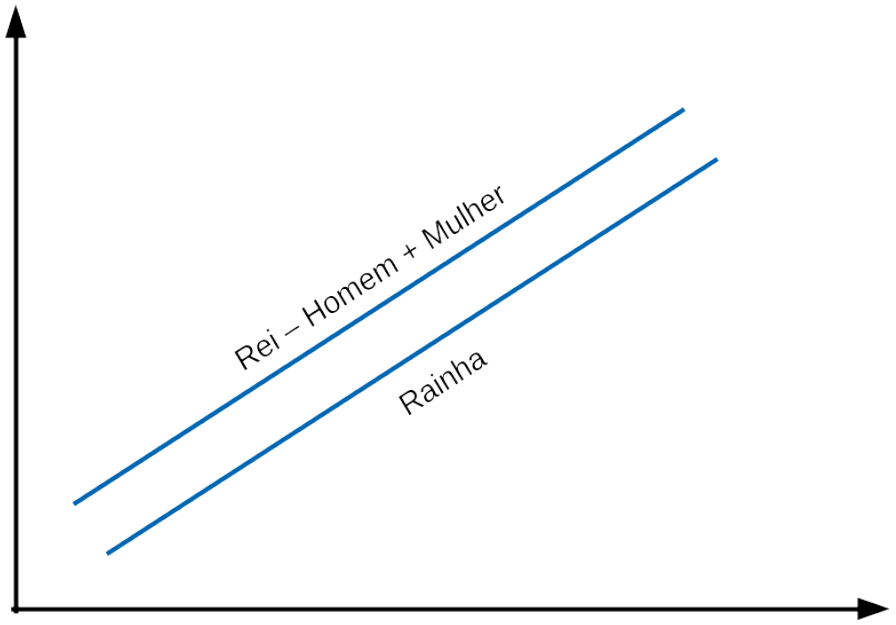
Word embeddings podem ser treinados para um domínio específico ou pode-se aproveitar modelos já treinados em grandes *corpora*. Modelos já treinados podem ser atualizados posteriormente. Alguns grupos de pesquisa na área de processamento de linguagem natural disponibilizam modelos já treinados. Para língua portuguesa, modelos treinados foram disponibilizados pelo grupo NILC [1]

2.4.3 Word2vec

O *Word2vec* é a implementação referencial em *word embedding*, sendo desenvolvido por Tomas Mikolov e outros pesquisadores da empresa Google e sendo publicado em 2013 (MIKOLOV et al., 2013a). Trata-se de um técnica para geração de representações distribuídas de palavras usando modelos de redes neurais. *Word2vec* representa as palavras em vetores numéricos distribuídos, aproximando as representações de palavras

[1] http://www.nilc.icmc.usp.br/embeddings

Figura 2.3: Composição Vetorial



Fonte: Os Autores

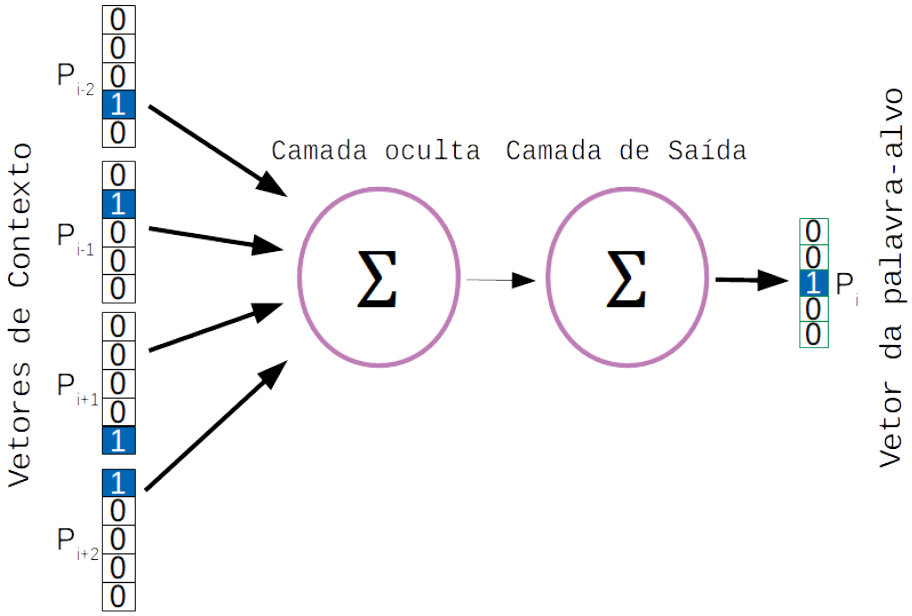
contextualmente similares.

A contribuição de Mikolov *et al.* com *word2vec* foi reduzir a complexidade computacional para aprender representação distribuídas de palavras, focando em aprendizado eficiente e possuindo apenas duas camadas internas, permitindo ser treinado em um volume de dados maior do que técnicas de *deep learning*. (GOLDBERG, 2017) Com isso, o algoritmo estabelece um bom compromisso entre qualidade da representação e custo computacional.

É possível utilizar dois modelos de arquitetura para produzir a distribuição da representação de palavras: o "*Continuous Bag of Words*" ou o *skipgram*. Ambos consistem em treinar uma rede neural de uma camada oculta e uma camada de saída para uma determinada tarefa, usando o algoritmo de otimização *stochastic gradient descent* e função custo derivada de *softmax*. O objetivo dos algoritmos não é usar as redes neurais para as tarefas treinadas, mas aprender os pesos da camada oculta e usá-los como representações distribuídas de palavras. Eles trabalham com o conceito de contexto ou janela de contexto que é definido como o conjunto de palavras que precedem e sucedem uma determinada palavra em um texto.

Para exemplificar o funcionamento dos dois modelos, suponha que o vocabulário seja composto por 100000 palavras e o algoritmo foi configurado para aprender vetores de palavras de 300 características. Nessa configuração a camada oculta consiste em uma matriz de pesos W_1 de 100000 linhas e 300 colunas e a camada de saída possui uma

Figura 2.4: Representação gráfica resumida do modelo *Continuous Bag of Words*



Fonte: Os Autores

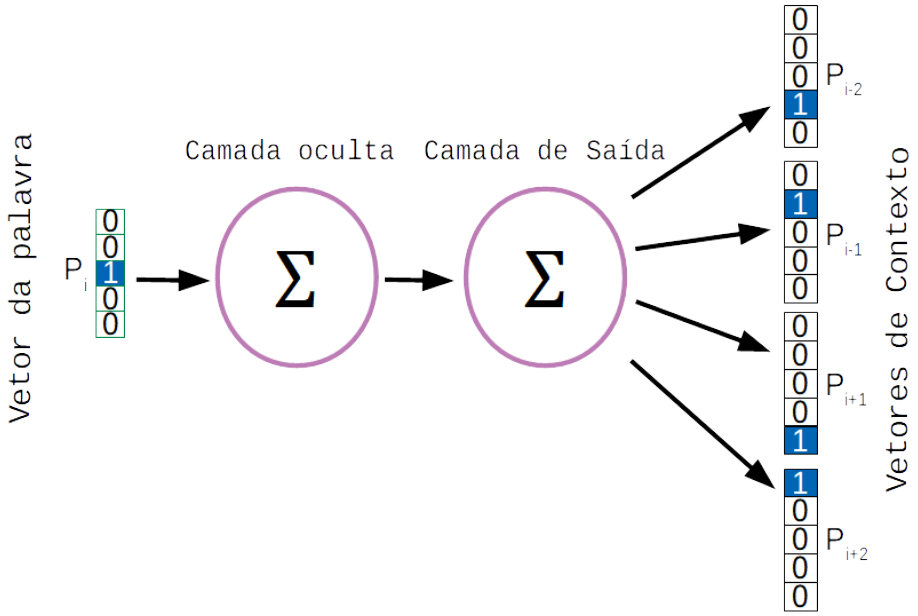
matriz de pesos W_2 de 300 linhas e 100000 colunas. As camadas de entrada e saída são compostas por uma palavra e seu contexto codificadas em vetores 1-de-n. Como os vetores de entrada usam codificação 1-de-n ao multiplicar o vetor de entrada pela matriz W_1, efetivamente selecionamos a linha correspondente à posição do valor um do vetor. Por isso, cada linha nessa matriz corresponde a uma representação distribuída de uma palavra no vocabulário.

Em *CBOW* busca-se prever uma palavra dado seu contexto. A camada de entrada é composta pelos vetores 1-de-n das palavras de contexto e a saída pelo vetor 1-de-n da palavra em foco. Por exemplo, dada uma entrada como P_{i-2}, P_{i-1}, P_{i+1}, P_{i+2}, onde P_i é o vetor 1-de-n da palavra na i-ésima posição do texto de entrada, o algoritmo objetiva maximizar a probabilidade de observar P_i na camada de saída. Esse exemplo pode ser visualizado na figura 2.4.

No caso de *CBOW*, a função de ativação da camada oculta, por padrão, consiste na média das linhas de W_1 correspondentes às palavras de contexto recebidas na camada de entrada. A saída da camada oculta é multiplicada pela matriz W_2 e a função softmax é aplicada para garantir que os elementos do vetor de saída sejam valores entre zero e um e a soma deles totalize exatamente um. O algoritmo varia os valores das matrizes de peso com objetivo que o vetor de saída convirja para a representação 1-de-n da palavra em foco.

A tarefa a ser resolvida pelo modelo *skip-gram* é a oposta do *CBOW*. Ele busca

Figura 2.5: Representação gráfica resumida do modelo *Skip-gram*



Fonte: Os Autores

prever a partir de uma palavra o seu contexto. A camada de entrada é composta pela palavra em foco e a saída pelo seu contexto. Na figura 2.5, temos o seguinte exemplo: dada uma entrada como P_i com o vetor 1-de-n da palavra na i-ésima posição do texto de entrada e uma janela de contexto de tamanho dois, o algoritmo objetiva maximizar a probabilidade de observar P_{i-2}, P_{i-1}, P_{i+1}, P_{i+2}, na camada de saída.

As camadas oculta e de saída são compostas pelas matrizes de pesos W_1 e W_2 no mesmo formato do modelo *CBOW*. A função de ativação da camada oculta consiste na seleção da linha da matriz de pesos correspondente à palavra em foco. Cada coluna da matriz W_2 corresponde a uma palavra de contexto. Na camada de saída é calcula a probabilidade de cada palavra do vocabulário estar no contexto da palavra em foco. Para se obter a probabilidade de uma palavra P_n estar no contexto de uma palavra P_m, multiplica-se a n-ésima linha da matriz W_1 pela m-ésima coluna da matriz W_2 e aplica-se a função softmax. O objetivo de treino é minimizar o somatório do erro de predição entre todos os vetores da camada de saída. Assim o modelo ajusta os pesos das duas matrizes de pesos de forma que a função softmax atribua a maior probabilidade de seleção às palavras de contexto observadas nos textos.

2.5 Representações de Documentos

Assim como palavras, representações numéricas eficientes de documentos podem facilitar tarefas processamento de linguagem natural como identificação de similaridade de documentos e classificação textual. Devido a sua simplicidade e eficiência, uma das representações vetoriais mais comuns é *bag-of-words*, literalmente traduzida como saco de palavras. Ela tem como ideia principal pensar em documentos como um conjunto de palavras desconsiderando a ordem dentro do contexto frasal (HARRIS, 1954). Sendo assim, a implementação mais simples de *bag-of-words* é um vetor binário com uma posição para cada palavra no vocabulário, onde apenas as posições correspondentes palavras presentes no texto são iguais a um. Isso contrasta com uma análise sintática mais complexa onde tenta-se inferir toda estrutura da frase ou dos parágrafos.

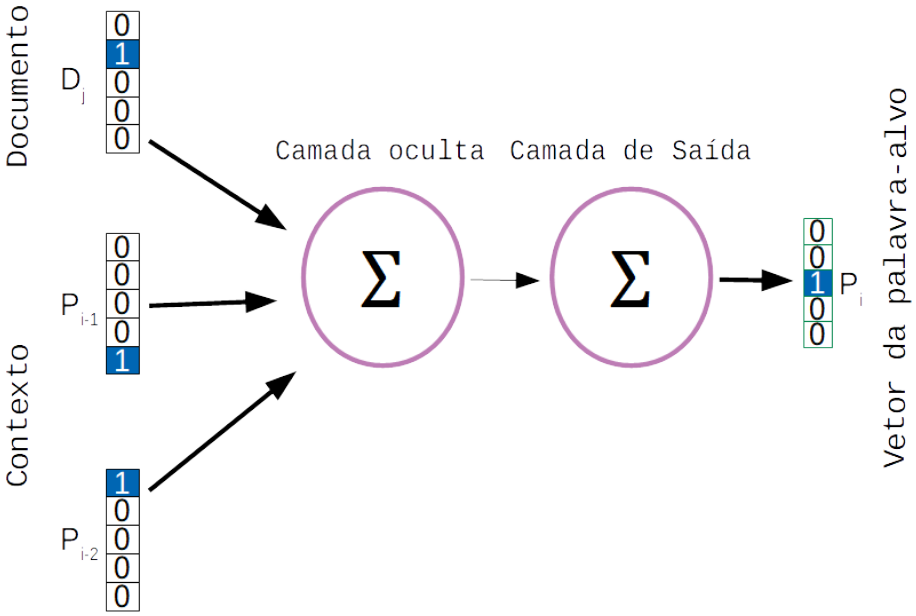
O modelo de *bag-of-words* possui às mesmas desvantagens de um representação 1-de-n para palavras: são representações esparsas, possuem muitas dimensões e pouca carga semântica. Nas seções 2.5.1 e 2.5.2 são apresentados os algoritmos Doc2Vec e FastText para geração de representações vetoriais para documentos densas e com propriedades semânticas.

2.5.1 Doc2Vec

Doc2Vec é uma implementação no framework Gensim (ŘEHŮŘEK; SOJKA, 2010) do conceito de *Paragraph Vector*, vetores de parágrafo em tradução livre, desenvolvido por Quoc Le e Tomas Mikolov (LE; MIKOLOV, 2014). O princípio de vetores de parágrafo é o análogo para documentos de *word embeddings*. Para obter essas representações, os autores propõem extensões aos modelos *Skip-gram* e *CBOW* da técnica *Word2vec*.

A primeira extensão é chamada PV-DM, sigla em inglês para Modelo de Memória Distribuída de Vetores de Parágrafos, e é baseada no modelo *CBOW*. Nela, em adição aos vetores de palavras de contexto, o algoritmo recebe como entrada um vetor 1-de-n que identifica o documento dentro do conjunto de teste. Além disso, a camada oculta possui uma matriz de pesos extra para representar os vetores distribuídos de documentos a serem aprendidos. A saída da camada oculta pode ser a função de ativação de concatenação ou média dos vetores distribuídos das palavras de entrada e do documento. Caso a função de ativação escolhida seja concatenação, a informação sobre a ordem das palavras no documento é preservada. O processamento na camada de saída ocorre da mesma forma

Figura 2.6: *Paragraph Vectors Distributed Memory*



Fonte: Os Autores

que o modelo *CBOW*.

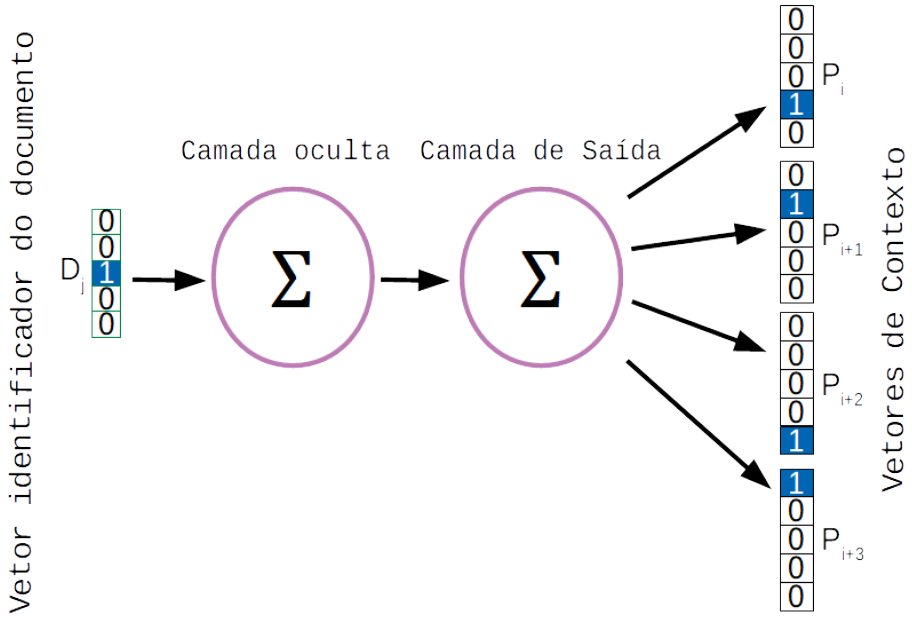
Por exemplo, dada uma entrada como D_j,P_{i-2}, P_{i-1}, onde D_j é o vetor 1-de-n do j-ésimo documento e P_i é o vetor da i-ésima palavra do documento, o algoritmo objetiva maximizar a probabilidade de observar P_i na camada de saída. Esse exemplo pode ser visualizado na figura 2.6.

A segunda extensão é chamada PV-DBOW, sigla em inglês para Modelo Distribuído de *Bag-Of-Words* para Vetores de Parágrafos, e é baseada no modelo *skip-gram*. Nela, a entrada é o vetor identificador do documento e a saída são os vetores 1-de-n de todas as palavras do documento. A camada oculta é composta por uma matriz de pesos das representações distribuídas dos documentos a serem aprendidos. O aprendizado ocorre da mesma forma que o modelo *CBOW*. Na figura 2.7, temos o seguinte exemplo: dada uma entrada como onde D_j é o vetor 1-de-n do j-ésimo documento com quatro palavras e P_i é a i-ésima palavra nesse documento, o algoritmo objetiva maximizar a probabilidade de observar P_i, P_{i+1}, P_{i+2} e P_{i+3},na camada de saída.

2.5.2 *FastText*

O algoritmo *fastText* foi desenvolvido por uma equipe de pesquisadores da empresa Facebook como um extensão do algoritmo *Word2vec*. Ele possui variantes para

Figura 2.7: *Distributed Bag of Words*



Fonte: Os Autores

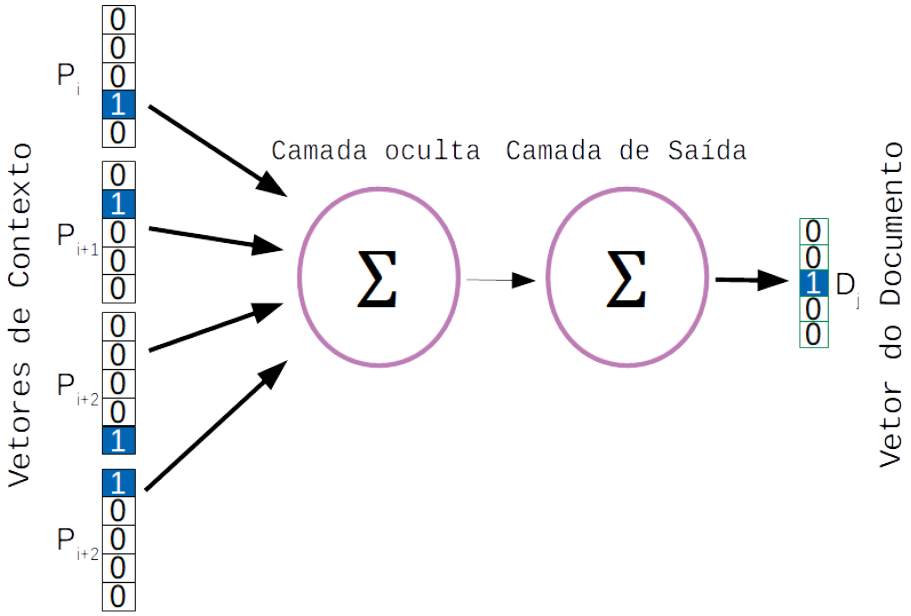
representação de palavras e documentos.

Para representação de palavras FastText baseia-se no modelo *skip-gram*. O algoritmo difere de *Word2Vec* no sentido que assume que palavras são compostas por n-gramas de seus caracteres (BOJANOWSKI et al., 2016). O algoritmo aprende representações vetoriais distribuídas para os n-gramas das palavras do vocabulário. A representação vetorial distribuída de uma palavra é a soma das representações de seus n-gramas.

Por exemplo, para 4-gramas, o algoritmo aprende as seguintes representações para a palavra rodovia: “<rod”, “rodo”, “odov”, “dovi”, “ovia”, “via>”, além da própria palavra representada por “<rodovia>”. Nesse exemplo, os caracteres “<” e “>” são usados para diferenciar prefixos e sufixos de outras sequências e caracteres. Essa abordagem possui algumas vantagens sobre *word2vec*: permite o aprendizado de derivações morfológicas, por exemplo, formas verbais; agrupar palavras com erros de digitação; tratar da composicionalidade da língua, no exemplo citado rodovia possui n-gramas em comum com rodoviária e via, palavras com as quais é relacionada; gerar representações melhores para palavras raras, pois seus n-gramas podem aparecer em mais contextos e aprendizado de palavras fora do vocabulário.

Para representação de documentos, *fastText* baseia-se no modelo *CBOW*. O algoritmo difere ao aplicar o conceito de *bag of n-grams*, ou seja, assume que documentos são compostos pelos n-gramas contendo suas palavras. Por exemplo, a frase “O trânsito está lento hoje” pode ser quebrada nos seguintes trigramas: “O trânsito está”, “trânsito

Figura 2.8: *fastText*



Fonte: Os Autores

está lento", "está lento hoje". A entrada do algoritmo são os vetores 1-de-n de todas os n-gramas do documento e a saída é o vetor identificador do documento. Pode-se pensar nessa abordagem como o inverso do modelo *PV-DBOW* apresentado na seção 2.5.1, como mostrado na figura 2.8.

3 TRABALHOS RELACIONADOS

Para o presente trabalho foram consultadas diversas obras que tratassem sobre os conceitos e sobre a relatos de experiência semelhantes ao que é proposto. Dessa forma, podemos apresentar primeiramente o trabalho de Wang *et al*: "*Traffic Condition Analysis Based on Users Emotion Tendency of Microblog*" (WANG et al., 2017), em que os autores propõem um modelo de extração de condição de tráfego baseado em análise de sentimento de usuários de microblogs. As postagens são transformadas em vetores usando a ferramenta do *Word2vec* para aumentar a representatividade dos termos que serão alimentados aos modelos. O modelo aplicado é semi-supervisionado, usando redes adversárias generativas condicionais (*conditional generative adversarial networks*) para incrementar o conjunto de teste a partir de dados de microblog anotados e utilizando *Gated Recurrent Unit* como classificador. Os autores concluem que foi obtido um melhor classificador de sentimentos ainda que há algumas melhorias que podem ser feitas por haver diferença entre os dados gerativos e dados de texto.

Nessa mesma linha, podemos pensar nos trabalhos de D'Andrea *et al*. (D'ANDREA et al., 2015), Sakaki *et al*. (SAKAKI et al., 2012), e Wanichayapong *et al*. (WANICHAYAPONG et al., 2011) que apesar de possuírem características específicas trabalham com a relação entre os *tweets* dos usuários e a detecção de incidentes de trânsito. Sendo assim, o primeiro trabalho "*Real-Time Detection of Traffic From Twitter Stream Analysis*" (D'ANDREA et al., 2015) os autores sugerem um sistema de monitoramento em tempo real para detecção de eventos de tráfego/trânsito. O sistema realiza buscas no Twitter, processa e classifica os *tweets* como relacionados a eventos de trânsito ou não. Para geração de vetores-chave (*feature vectors*) o trabalho reduz as frases a stems mais relevantes por meio da métrica de ganho de informação e atribui a cada uma um peso correspondente ao IDF (*inverse document frequency*). O trabalho avalia diversos modelos de classificação e conclui que SVMs obtêm o melhor resultado, visto que eles obtiveram uma *F1-score* de 95,75% em problemas de classificação binária e de 88,89% em problemas de classificação ternária, nos quais os autores também consideraram os eventos externos ao tráfego. Os autores apontam também a capacidade do sistema de antecipar a capacidade do sistema de antecipar a detecção em relação aos meios oficiais de comunicação do país.

Já em "*Real-time Event Extraction for Driving Information from Social Sensors*", de Sakaki *et al.* (SAKAKI et al., 2012) as redes sociais são utilizadas para extração de eventos de trânsito. Para obter essas postagens, eles filtram *tweets* por meio de termos-

chave pré-definidos. As amostras são rotuladas em "informação de trânsito" ou "não-informação de trânsito" e o classificador é treinado usando SVM. Por fim, a localização das postagens foi classificada como "informação de trânsito" por meio de informações de GPS e usando técnicas de dicionário para identificação de entidades nomeadas. Essa é uma proposta bem interessante, justamente por permitir mapear os *tweets* o que pode se desdobrar em um trabalho e/ou um dispositivo que atualize em tempo real os incidentes de trânsito de forma a organizar sistematicamente os eventos e disponibilização para o público. Os resultados que os autores apresentam definem como sendo uma nova abordagem utilizada para a extração de informações das mídias sociais e que com o sistema de demonstração que pode coletar *tweets* referentes a tráfego intenso no Twitter com precisão de cerca de 87% e pode extrair informações de localização com precisão de 85%.

Ainda sobre a extração de informação de trânsito, Wanichayapong *et al* em "*Social-based Traffic Information Extraction and Classification*" (WANICHAYAPONG et al., 2011) classificam *tweets* de trânsito por meio de análise sintática. Esses são extraídos por meio de buscas na API do Twitter usando palavras-chave pré-determinadas. O sistema é capaz de identificar eventos pontuais e em trechos, bem como posicioná-los geograficamente. Os autores aplicam técnicas de análise sintática para extrair e classificar as postagens relacionadas a tráfego e ainda utilizando técnicas de tokenização e ressaltam a importância de dicionários para aumentar a precisão do algorítimo. Outro fator importante é o geocodificação que auxilia a lidar com lugares com nomes similares. Mas, conforme conclui o artigo ainda é necessário refinar o classificador para alguns contextos adjacentes.

Os autores Schulz *et al*. combinam web-semântica e aprendizado de máquina para detectar informações advindas de redes sociais relacionadas a incidentes de pequena escala em "*I See a Car Crash: Real-Time Detection of Small Scale Incidents*" (SCHULZ; RISTOSKI; PAULHEIM, 2013). O trabalho propõe etapas de coleta, pré-processamento, extração de atributos, treinamento e detecção em tempo real. A primeira inclui propostas de filtragem de re-*tweets*, remoção de *stopwords*, expansão de abreviaturas, correção de erros ortográficos, substituição de marcadores temporais e espaciais, lematização, POS-tagger e *filter*. Abordagem com diversos atributos, como estatísticos, sintáticos e semânticos usando Linked open data. Três classificadores foram testados no trabalho: Naïve Bayes Binary Model (NBB), Ripper ruler learner (JRIP) e SVM. A última etapa processa os *tweets* usando HeidelTime, um reconhecedor de expressões temporais (*temporal tagger*), e Standford NER para extrair essas informações e GeoNames para resolver entidades

nomeadas para geolocalização. O sistema é verificado comparando os eventos identificados por ele com os reportados por entidades do governo responsáveis pela gerência do trânsito.

Um trabalho que apresenta especificamente uma metodologia para adquirir, processa e filtrar *tweets* relacionados a incidentes de trânsito, temos o trabalho de Gu *et al*. que em "*From Twitter to detector: Real-time traffic incident detection using social media data*" (GU; QIAN; CHEN, 2016). Os *tweets* de incidentes de trânsito são adquiridos por meio da API REST do Twitter em tempo real usando um dicionário de palavras-chaves relevantes e suas combinações. Cada tweet é mapeado para um vetor binário multidimensional no espaço formado pelo dicionário, o que recorre a ideia de *Adaptive data acquisition*. Os *tweets* são classificados em incidente de trânsito (TI) e não-incidente (NTI) usando Semi-Naive-Bayes. Cada *tweet* classificado como TI é classificado por *Supervised Latent Dirichl location* para identificar a categoria do incidente. Além disso, os autores utilizam três categorias: *clustering-based, model-based e signal-processing-based*. Os autores concluíram que uma pequena amostra do montante adquirido cobria a maioria dos incidentes relatados no conjunto dos dados. Além disso, foi constatado que um *tweet* útil que fosse relacionado a acidentes de trânsito, representava aproximadamente 5% de todos os *tweets* adquiridos e que desses, de 60 a 70% eram postados por usuários influentes, ou seja, contas públicas do Twitter que na maioria das vezes pertenciam a órgãos públicos e mídia, enquanto o restante é contribuído por usuários individuais.

Outro diferencial dessas abordagens é o trabalho com grandes arquivos (*big data*) a partir das postagens disponíveis no Twitter. Saindo dessa única fonte de informações, é possível encontrar trabalhos que utilizam a plataforma Sina Weibo [1], rede social chinesa da empresa Sina Corporation em 2009 e que oferece serviço de microblogging semelhante ao Twitter.

Desses trabalhos, citamos "*Extraction of Traffic Information from Social Media Interactions: Methods and Experiments*" de Cui *et al*.(CUI et al., 2014) em que os autores propõem um método que alia extração de informações sobre tráfego compartilhadas autonomamente nessa plataforma com contribuições explícitas de usuários para melhorar a qualidade das informações extraídas. O sistema proposto foi composto de uma plataforma cliente-servidor. O servidor é responsável por coletar publicações na plataforma Sina Weibo, identificar aquelas relacionadas a incidentes de trânsito e estruturar as informações dos eventos de trânsito identificando o tipo de incidente, localização e horário.

[1] https://www.weibo.com/signup/mobile.php?lang=en-us

Um classificador Bayesiano foi usado sobre os bigramas das publicações para separá-las entre não-relacionadas a trânsito e relacionadas a trânsito, sendo essa última categoria posteriormente dividida em três: incidente de trânsito, fluxo de tráfego e localização. O foco principal desse trabalho era voltado para o preparo de informações de tráfego para o caso de tratamento de emergências.

Por sua vez, Chen at al. apresentam em "*A Convolutional Neural Network for Traffic Information Sensing from Social Media Text*" (CHEN et al., 2017), um método de extração de microblogs com informações relevantes para tráfego da plataforma Sina Weibo. Eles utilizaram um modelo de "*continuous bag-of-words*" para aprender as representações baseadas em vetores de palavras a partir de 3 bilhões de postagens no microblog não rotulados. Essas representações são usadas para treinar uma rede neural convolucional (CNN) em conjunto com uma série de postagens rotuladas como relacionadas a trânsito ou não. Os autores propõem integrar os dados da mídia social e dos detectores de tráfegos físicos tradicionais como uma forma atrativa de lidar com o tópico do transporte.

Com base nesses trabalhos podemos pensar na metodologia para a produção desse trabalho usando o exemplo dos trabalhos já feitos e considerando o referencial teórico elencado nesse capítulo.

4 MATERIAIS E MÉTODOS

O trabalho consiste na criação de um classificador de *tweets*, para identificação de incidentes de trânsito. O aprendizado é supervisionado, sendo feita a classificação manual deles em duas categorias: "trânsito" e "não trânsito". Após isso, esses *tweets* foram utilizados para treinar e testar os modelos *Doc2Vec* e *fastText*, comparando o desempenho de ambos para a automatização da categorização de textos de trânsito.

Esse capítulo descreve a metodologia utilizada para a busca, seleção, extração, identificação dos *tweets* de Trânsito e posteriores validações realizadas. Na seção 4.1 serão apresentadas o hardware e software da máquina utilizada. Na seções 4.2.2 e 4.2.3 os critérios de seleção inicial para identificar os *tweets* relevantes para o trabalho. Na seção 4.2.4 serão apresentados os critérios para classificação manual, utilizando alguns conceitos de Processamento de Linguagem Natural como relevantes para essa etapa. Na seção 4.2.5 é definido o aprendizado supervisionado e descrita a metodologia de avalidação.

4.1 Hardware e Software Utilizado

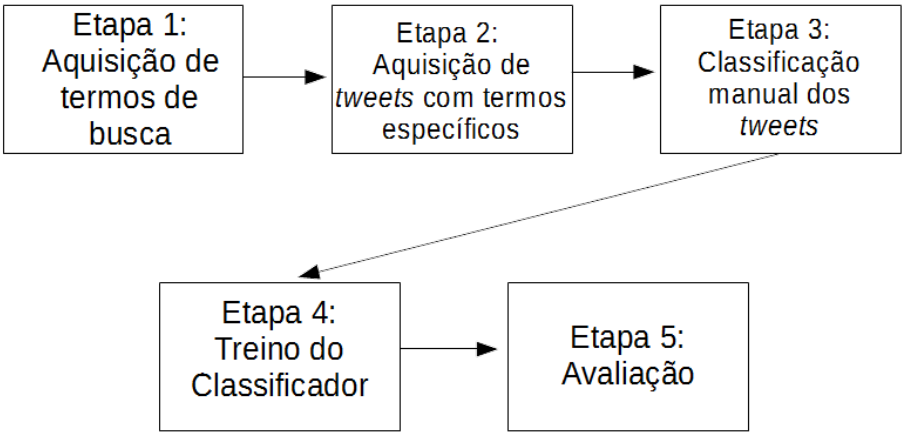
Para todos os testes executados foram utilizados o mesmo computador, sendo que este possui as seguintes características de hardware: Processador Intel 4250U de 64 bits com frequência base de 1,3 GHz, 2 núcleos de processamento e 4 *threads*, 4GB de memória RAM com frequência de 1600MHz.

O sistema operacional utilizado foi Mac OS X 10.13 de 64 bits. As seguintes ferramentas de *software* foram usadas:

- Linguagem - Python.[1]
- Pré-processamento - bilioteca NLTK (*Natural Language Tool Kit*)[2] em conjunto com Unidecode.[3]
- Doc2Vec - biblioteca Gensim.[4]
- fastText - interface Python no repositório oficial.[5]

[1] https://www.python.org/
[2] http://www.nltk.org/
[3] https://pypi.org/project/Unidecode/
[4] https://radimrehurek.com/gensim/
[5] https://github.com/facebookresearch/fastText/tree/master/python

Figura 4.1: Etapas do Desenvolvimento do Trabalho



Fonte: Os Autores

- Webscraper do Twitter - biblioteca Twitterscraper.[6]

4.2 Desenvolvimento

O trabalho foi desenvolvido a partir das seguintes etapas, como mostra figura 4.1:

1. Aquisição de termos de busca
2. Aquisição de *tweets* candidatos
3. Classificação manual dos *tweets*
4. Treinamento do classificador
5. Avaliação do classificador

As etapas de 1 à 4 serão explicadas nas seções 4.2.2 à 4.2.5. A etapa 5 sobre avaliação será explicada e seus resultados apresentados no capítulo 5. O passo de pré-processamento está presente nas etapas 1 e 4, por isso ele será apresentado separadamente na seção 4.2.1.

4.2.1 Pré-processamento

A tarefa de pré-processamento do texto é muito relevante para a qualidade dos resultados em classificação textual (UYSAL; GUNAL, 2014). Cinco técnicas de pré-

[6]https://github.com/taspinar/twitterscraper

processamento foram aplicadas neste trabalho. Elas são apresentadas na lista a seguir, sendo que as duas primeiras sempre ocorrem, enquanto aplicação de cada uma das três últimas é um parâmetro configurável do programa:

Análise léxica e Sanitização - Uma divisão do texto corrido em termos definidos. Esse passo identifica por meio de expressões regulares palavras, nomes, números, datas, horários e emojis e remove qualquer outra cadeia de caracteres. Por exemplo, a entrada "00:57 - Blitz na Salvador França, sentido bairro <-> centro."é dividida em: "00:57", "Blitz", "na", "Salvador", "França", "sentido", "bairro"e "centro", removendo toda pontuação e o símbolo «->".

Normalização da entrada - Todas as palavras foram convertidas para letras minúsculas, para evitar que, por exemplo, “Porto Alegre” escrito com as iniciais em maiúscula, fosse visto como um termo diferente de “porto alegre”. Também foi escolhido remover qualquer forma de acentuação para garantir que alguns erros na grafia na linguagem informal fossem relevados. Para isso, foram mapeados os caracteres da codificação *unicode* para os equivalentes em *ASCII* usando a biblioteca *Unidecode*. Por exemplo, a entrada "00:57", "Blitz", "na", "Salvador", "França", "sentido", "bairro"e "centro"torna-se: "00:57", "blitz", "na", "salvador", "franca", "sentido", "bairro"e "centro".

Remoção de *stopwords* - *Stopwords* ou “palavras vazias” são termos comuns da língua que usualmente apresentam pouca relevância semântica. São palavras necessárias na linguagem natural por questões sintáticas, mas que podem ser retiradas de frase que a mesma sem perda de valor semântico. São exemplos: artigos, conjunções, preposições, interjeições, verbos auxiliares e palavras muito comuns. Foram usadas ferramentas da biblioteca *NLTK* para remoção inicial de *stopwords*. Por exemplo, a entrada "00:57", "blitz", "na", "salvador", "franca", "sentido", "bairro"e "centro"torna-se: "00:57", "blitz", "salvador", "franca", "sentido", "bairro"e "centro", pois a palavra "na"é considerada uma *stopword*.

Stemming - A técnica de *stemming* foi escolhida para melhorar o contexto das palavras, pois reduz diversas derivações morfológicas para uma mesma raiz. Foi escolhida a implementação do algoritmo RSLP (Removedor de Sufixo da Língua Portuguesa) disponível na biblioteca NLTK. (ORENGO; HUYCK, 2001) Por exemplo, a entrada "00:57", "blitz", "salvador", "franca", "sentido", "bairro"e "centro"torna-se: "00:57", "blitz", "salv", "franc", "sent", "bairro"e "centr".

Agrupamento de classes de termos - Algumas classes termos representam propriedades relevantes para incidentes de trânsito. Números podem representar duração ou extensão de um congestionamento. Datas e horários auxiliam a posicionar os incidentes no tempo. Contudo, ao manter esses dados na sua forma original durante a etapa de classificação, as postagens sendo analisadas apresentam maior variabilidade, devido às diversas combinações possíveis para esses itens. Para mitigar essa situação, para cada uma dessas classes foi criado um marcador. Cada marcador segue o formato "__NUMERO__", "__DATA__" e "__HORARIO__", garantindo que nenhum deles represente uma palavra do vocabulário. Por exemplo, a entrada "00:57", "blitz", "na", "salvador", "franca", "sentido", "bairro"e "centro"torna-se: "__HORARIO__", "blitz", "na", "salvador", "franca", "sentido", "bairro"e "centro", com o termo "00:57"sendo substituído por "__HORARIO__".

4.2.2 Aquisição de termos de busca

Um dos objetivos desse trabalho é desenvolver uma ferramenta que permita identificar *tweets* que indiquem incidentes de trânsito em Porto Alegre. Os *tweets*, em geral, tratam de assuntos diversos sendo que a maior parte deles não são relevantes para identificar incidentes de trânsito. Devido ao grande volume de *tweets* produzidos constantemente é inviável tentar classificar cada um deles. Como solução, o *Twitter* permite filtrar as postagens mais recentes por conteúdo e outras propriedades. Para usar essa funcionalidade, é necessário definir termos de busca relevantes. Os termos relevantes foram adquiridos analisando publicações de contas de usuário especializadas em publicação de incidente de trânsito em Porto Alegre. Nesse trabalho foram escolhidas as contas: @EPTC, da Empresa Pública de Transportes e Circulação, e @MarinaPagno, responsável pelas publicações de trânsito do Grupo Zero Hora. A partir dessas publicações foram levantadas as principais expressões usadas e gerada uma lista de termos de busca.

As expressões foram selecionadas processando o texto, agrupando as palavras em unigramas e bigramas e ordenando-os pelo número de ocorrências. Foram escolhidos os 400 termos mais comuns e todo unigrama que apareceu dentro de um bigrama foi removido da lista, resultando em 377 termos. Isso permite a evitar que, por exemplo, "porto" e "alegre" aparecessem isolados de "porto alegre", pois o bigrama representa uma expressão mais específica e expressiva. Bigramas frequentemente representavam estruturas mais gramaticais, como substantivo e adjetivo, por exemplo "trânsito lento". A ferramenta de

busca do *Twitter* permite buscas de termos não adjacentes, ou seja, colocando duas entradas “trânsito” e “lento”, ele pega expressões como “trânsito muito lento”; “trânsito um pouco lento” e “acidente de trânsito deixa fluxo lento”. Trigramas e agrupamentos maiores já se apresentavam como muito específicos, fazendo com que houvessem menos ocorrências.

A API do Twitter tem uma limitação no número de acessos que podem ser feitos em um determinado período.[7] Com isso, para agilizar o processo e retirar a limitação de acesso foi utilizada a técnica de *web scraping* para extrair *tweets* a partir da interface web de busca. *Web scraping* envolve realizar requisições para um servidor web, simulando o acesso de um usuário; analisar e estruturar os dados recebidos. Foi decidido não implementar por conta própria um *web scraper* para *Twitter*, aproveitando a ferramenta *TwitterScraper*. Essa ferramenta permite usar os mesmos argumentos da interface de busca web.

4.2.3 Aquisição de *tweets* candidatos

Os termos de busca selecionados foram utilizados para coletar *tweets* originados de usuários de Porto Alegre usando o mesmo *web scraper* da etapa anterior. Os *tweets* foram coletados em ordem cronológica inversa de seis meses a partir do dia 23 de março de 2018. Por questões de privacidade do usuário apenas o conteúdo do *tweet* foi armazenado. Informações de metadados, como nome de usuário e geolocalização foram descartadas. Eles foram armazenados com conteúdo integral em arquivos de texto individuais e agrupados em pastas conforme o termo de busca usado para obtê-los.

4.2.4 Classificação manual dos *tweets*

Os *tweets* foram classificados manualmente em duas classes: “trânsito” e “não trânsito”. Pertencem a primeira classe apenas *tweets* que reportam sobre incidentes ou a situação do trânsito em Porto Alegre. Essa classe inclui reclamações sobre lentidão ou melhora das condições de trânsito, informes sobre acidentes, desvios e quaisquer outros eventos que estivessem afetando o trânsito, como alagamentos, neblina, obras e concertos dentre outros. Qualquer *tweet* que não pertença a classe “trânsito” é atribuído à “não

[7] https://developer.twitter.com/en/docs/basics/rate-limiting.html

trânsito". O processo de classificação manual foi subjetivo e envolveu duas pessoas. Ele requeria consenso para classificar cada *tweet* para reduzir vieses pessoais. Para cada termo de busca, os arquivos dos *tweets* foram separados em pastas conforme a classe atribuída. Deste montante, os *tweets* de 97 termos foram desconsiderados, por não retornarem apenas conteúdo automatizado de outras mídias sociais ou aparentemente aleatório que não contribuiria para o aprendizado dos classificadores. Ao fim desta etapa, foram classificados 10.351 *tweets*. Destes, foram atribuídos 4165 à classe "trânsito" e 6186 à "não trânsito".

4.2.5 Treino dos classificadores

Uma vez que os arquivos que representam os *tweets* estejam organizados conforme descrito na seção 4.2.4, o programa carrega os arquivos e os divide em conjunto de treino e teste. O primeiro conjunto é utilizado para treinar o modelo de aprendizado e o segundo para validar o aprendizado do mesmo. O conjunto de treino é balanceado por termo de busca, ou seja, para cada termo de busca são contados quantos *tweets* estão classificados como trânsito ou não-trânsito e são coletados aleatoriamente de cada grupo uma quantia igual ao menor grupo. Com isso, evitamos o viés de termos que possuem muitas instâncias de uma classe.

Para o conjunto de treino foram utilizados 2.234 *tweets*, sendo 1.117 de cada classe. O restante dos *tweets*, totalizando 8.117, foram usados para avaliação. Ao carregar os *tweets* são executados os passos de pré-processamento obrigatórios descritos na seção 4.2.1.

As implementações de vetores de parágrafos na biblioteca *Gensim* e o modelo de classificação textual *fastText* permitem identificadores arbitrários de documento. Com isso, podemos atribuir, além do identificador único, um rótulo para a classe que pertence o documento em questão. Ao final do treinamento temos representações vetoriais distribuídas das classes, além das representações dos documentos. Isso permite calcular a similaridade dos vetores de novos documentos com os vetores de classe. O classificador atribui o documento à classe com representação mais similar.

5 RESULTADOS E AVALIAÇÃO

5.1 Metodologia de Avaliação

Os classificadores foram analisados com os modelos treinados sobre o conjunto de treino e testados contra o conjunto de teste produzidos conforme descrito na seção 4.2.5. A análise compara o desempenho dos classificadores segundo a métricas *F1-score*, que consiste na média harmônica da precisão e revocação da classificação, bem como tempo de duração do treinamento. Também foi analisado o impacto dos parâmetros configuráveis de cada modelo e das técnicas de pré-processamento descritas na seção 4.2.1.

Os classificadores possuem os seguintes parâmetros configuráveis em comum que podem influenciar na qualidade dos resultados:

- *epochs* - Taxa de aprendizado
- *learnRate* - Taxa de aprendizado
- *learnDropRate* - Taxa de atualização de aprendizado
- *lossFunction* - Função de custo
- *minCount* - Número mínimo de ocorrências de um termo
- *sampleThreshold* - Função de custo
- *vecSize* - Tamanho do vetor dos características
- *winSize* - Tamanho da janela de contexto

Além desses, *fastText* possui os seguintes parâmetros:

- *ngrams* - Número de caracteres nos n-grams das palavras
- *wordGrams* - Número de palavras nos *bag of n-grams* dos documentos

Enquanto *Doc2Vec* possui o seguinte parâmetro particular:

- *trainingAlgorithm* - Modelo de treino do algoritmo

Os valores iniciais dos parâmetros dos algoritmos foram baseados em configurações sugeridas nos artigos que os apresentam e documentação oficial disponível online,

bem como orientações em fóruns oficiais de usuários.[1][2][3][4] A tabela 5.1 contém as configurações iniciais dos classificadores.

Tabela 5.1: Configurações iniciais dos parâmetros

Parâmetro	*Doc2Vec*	*fastText*
epochs	20	20
learnRate	0.05	0.05
learnDropRate	100	100
lossFunction	-5	1
minCount	5	5
sampleThreshold	0.01	0.001
vecSize	50	50
winSize	5	5
trainingAlgorithm	0	-
ngrams	-	0
wordGrams	-	1

Uma vez definidos os parâmetros iniciais, a contribuição de cada parâmetro foi avaliada individualmente. Para isso, o valor do parâmetro sendo analisado foi alterado dentro de sua faixa válida enquanto os outros parâmetros foram mantidos fixos. Os parâmetros em comum entre os classificadores foram alterados na mesma faixa de valores. A tabela 5.2 contém as faixas de valores analisadas para cada parâmetro.

Tabela 5.2: Configurações iniciais dos parâmetros

Parâmetro	*Valor*			
epochs	20	40	80	250
learnRate	0.05	0.1	0.25	0.5
learnDropRate	50	100	200	
lossFunction	-5	-1	0	1
minCount	0	1	5	
sampleThreshold	0.01	0.001	0.0001	
vecSize	25	50	100	300
winSize	3	5	9	
trainingAlgorithm	0	1	2	3
ngrams	0	5	6	
wordGrams	1	3	5	

Além desses parâmetros, o programa permite escolher alguns passos extras de pré-processamento do texto, conforme definido na seção 4.2.1. Foram analisadas todas as combinações das seguintes técnicas de pré-processamento: remoção de *stopwords*,

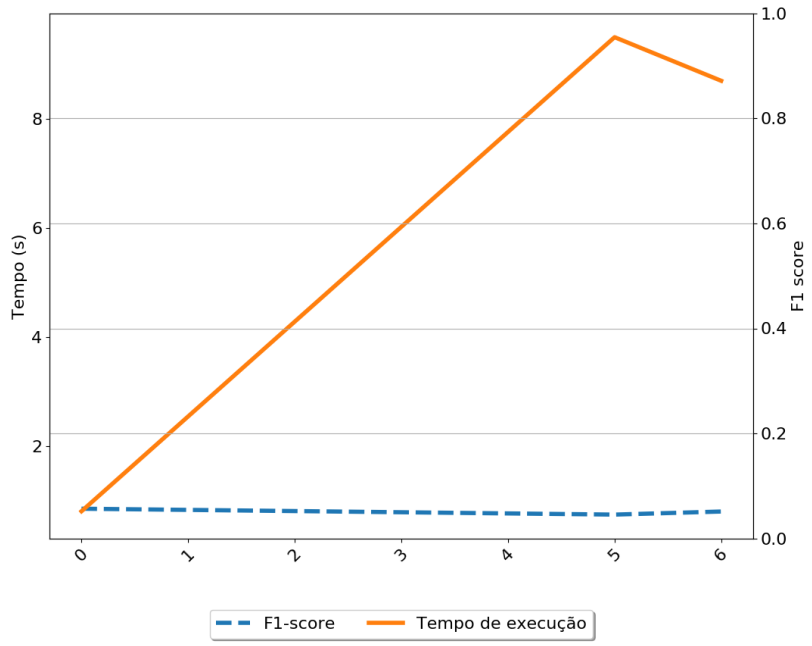
[1]Documentação FastText: https://fasttext.cc/docs/en/support.html
[2]Fórum FastText: https://groups.google.com/forum/#!forum/fasttext-library
[3]Documentação Word2Vec: https://code.google.com/archive/p/word2vec/
[4]Fórum Gensim: https://groups.google.com/forum/#!forum/gensim

stemming e agrupamento de classes de termos. A melhor combinação de técnicas de pré-processamento para cada um dos dois modelos foi escolhida como configuração fixa para análise dos outros parâmetros.

5.1.1 Análise de Configurações Individuais

FastText possui três contribuições para classificação de texto em relação a *Doc2Vec*: uso de n-gramas de caracteres das palavras, uso de *bag of n-grams* de palavras do texto e seu modelo baseado em *CBOW*. Contudo, como pode ser visto nas figuras 5.1 e 5.2, os parâmetros *ngrams* e *wordgrams*, correspondentes à primeira e à segunda contribuições, apresentam melhores resultados tanto em tempo quanto em *F1-score* quando configurados para os valores 0 e 1, respectivamente, que correspondem a não usar essas técnicas. Esses resultados provavelmente se devem às características do texto de um *tweet*: textos curtos, com gírias e expressões específicas da mídia analisada aumentam a variabilidade das palavras e reduzem as a quantidade de palavras de contexto entre as amostras.

Figura 5.1: FastText - *n*gramas

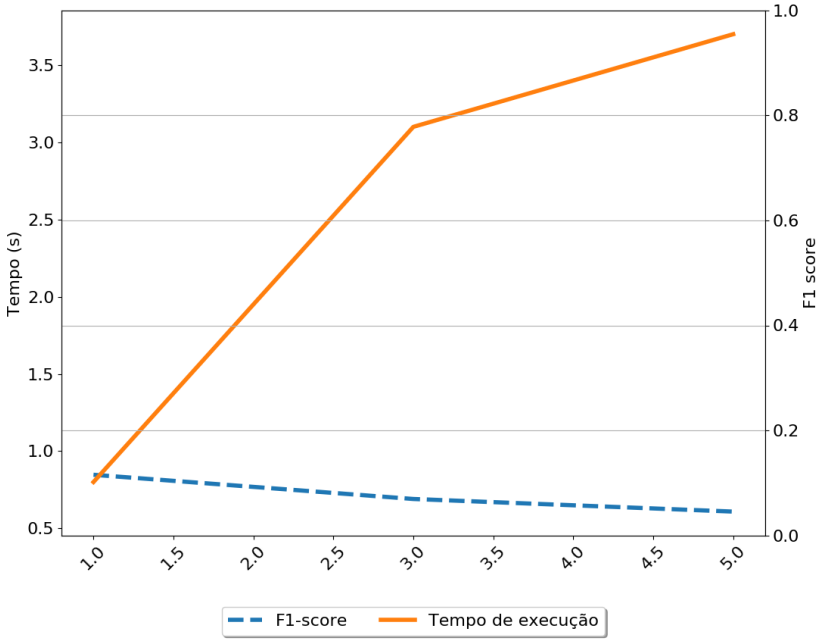


Fonte: Os Autores

Doc2Vec possui quatro opções de algoritmo de treinamento:

0. *PV-CBOW*
1. *PV-DM* com concatenação de vetores

Figura 5.2: FastText - *WordGrams*



Fonte: Os Autores

2. *PV-DM* com soma de vetores

3. *PV-DM* com média de vetores

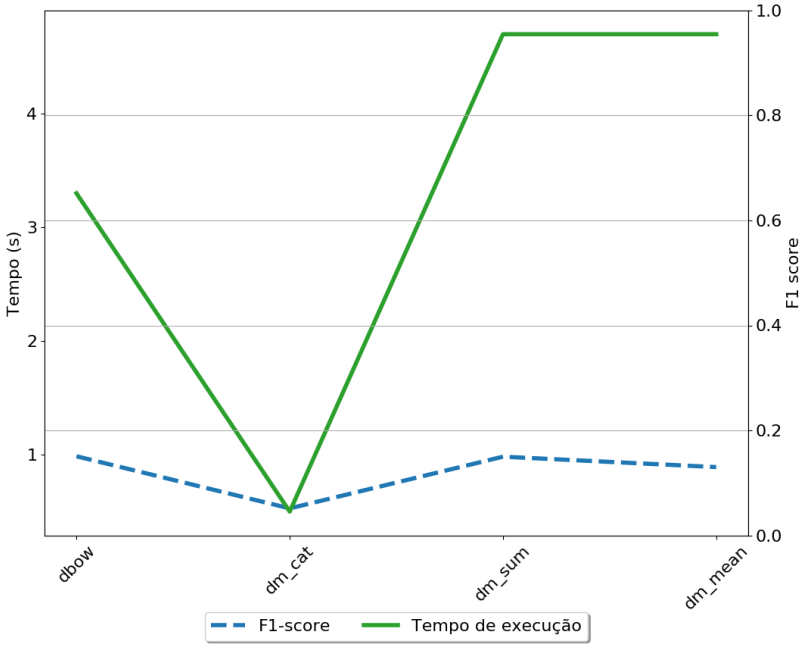
Como pode ser visto na figura 5.3, os algoritmos apresentam uma relação quase linear entre custo computacional, representado pelo tempo de execução do treinamento, e qualidade da classificação, representado pela métrica *F1-score*. O algoritmo *CBOW* estabelece o melhor compromisso entre qualidade e desempenho. Ele possui os melhor valor de *F1-score*, ambos acima de 98% e o segundo menor tempo de execução. Como já Le e Mikolov, o algoritmo *CBOW* possui um custo muito menor do que *PV-DM* devido a utilizar apenas uma matriz de pesos e aprender apenas os vetores dos documentos.(LE; MIKOLOV, 2014) A vantagem da técnica *PV-DM* aparece em problemas que requeiram conhecimento sobre a ordem das palavras, informação perdida *CBOW*.

5.1.2 Análise dos Parâmetros de Modelo e Pré-processamento

Esta seção compara cada parâmetro de modelo e combinação de opções de pré-processamento apresentados na seção 5.1.

O primeiro parâmetro analisado é a taxa de aprendizado (*learnRate*). Esse parâmetro define o quanto os pesos das matrizes das camadas oculta e de saída mudam a cada

Figura 5.3: Doc2Vec - Algoritmo de Treino

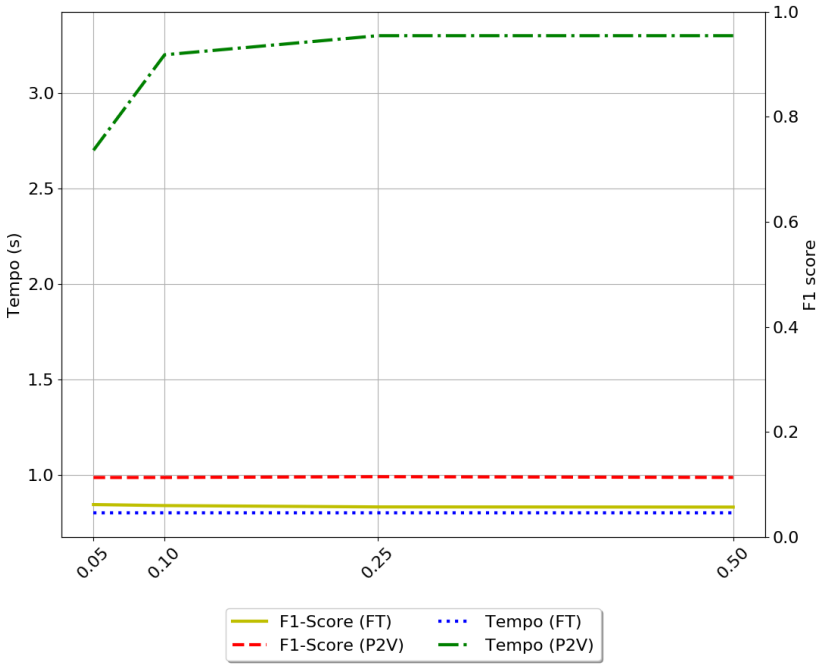


Fonte: Os Autores

iteração do algoritmo. Sendo assim, ela controla o quão rápido o modelo é atualizado durante o treinamento. Quanto maior o resultado, mais rápido o modelo converge para uma solução, contudo com o risco de sobre-ajuste para o conjunto de treino. A taxa de aprendizado quando mantida na faixa analisada entre 0,05 e 0,5 não afeta fortemente os resultados de ambos algoritmos. *FastText* é mais sensível a esse parâmetro, mostrando uma relação inversa entre *F1-score* e taxa de aprendizado, contudo a variação é menor do que 2%, como pode ser visto na figura 5.4. Quanto ao custo computacional, *Doc2Vec* requer mais tempo para o treino e parece ser afetado pelos parâmetros. Contudo a variação é de apenas 20% entre o melhor valor, 2,7s, e o pior, 3,24s.

O parâmetro *epochs* define quantas iterações o algoritmo executa sobre o conjunto de dados. Conjuntos de dados menores geralmente requerem mais iterações para obter representações de qualidade. Em uma análise prévia, definimos 50 iterações como valor inicial. Como pode ser visto na figura 5.5, os melhores resultados se encontram na faixa entre 20 e 40 iterações. Mesmo com um conjunto de dados relativamente pequeno quando comparado os usados nos trabalhos que apresentam os algoritmos, o modelo treinado neste trabalho consegue convergir para um modelo de qualidade com um número relativamente pequeno de iterações. A convergência rápida é muito importante, pois o número de iterações possui uma relação linear com o custo computacional. Cabe destacar que *fastText* possui treinamento mais eficiente em relação a *Doc2Vec*: cada iteração dura

Figura 5.4: Taxa de aprendizado



Fonte: Os Autores

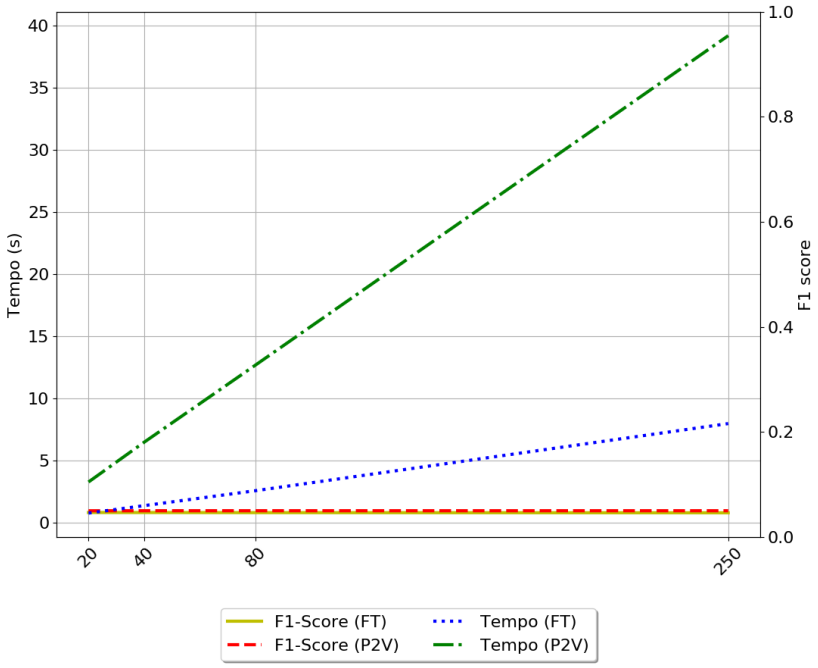
em média 0,03s contra 0,15s.

O tamanho do vetor (*vecSize*) controla quantos elementos os vetores distribuídos que representam os documentos e as palavras possuem. Quanto mais elementos, mais informações podem ser capturadas, mas também mais treinamento é necessário. Como pode ser visto na figura 5.6, esse parâmetro influencia na *F1-score* pouco dentro da faixa escolhida para análise. Assim como o número de iterações, o tamanho do vetor possui uma relação linear com o custo computacional, porém com menor contribuição.

O parâmetro *minCount* representa o número mínimo de ocorrências de uma palavra para ela ser considerada parte do vocabulário. Esse parâmetro ajuda a remover ruído do modelo, contudo caso um valor seja muito alto os resultados podem ficar enviesados para as palavras mais comuns. O número mínimo de ocorrências é outro parâmetro que apresentou pouco impacto na qualidade do modelo na faixa de valores analisada. A figura 5.7 mostra o comportamento inverso na *F1-score* entre *fastText* e *Doc2Vec* na variação do parâmetro. Novamente, o custo computacional de *fastText* é inferior e não varia com o parâmetro, enquanto *Doc2Vec* apresenta uma pequena variação de pouco mais de 20%.

O parâmetro *winSize* representa o número de palavras escolhidas para formar a janela de contexto em cada amostra de cada documento. Janelas de contexto com valores muito baixos podem levar a poucas coincidências entre documentos e sub-ajuste do modelo. Enquanto valores elevados podem causar o efeito inverso, sobre-ajuste. Taxa

Figura 5.5: Iterações



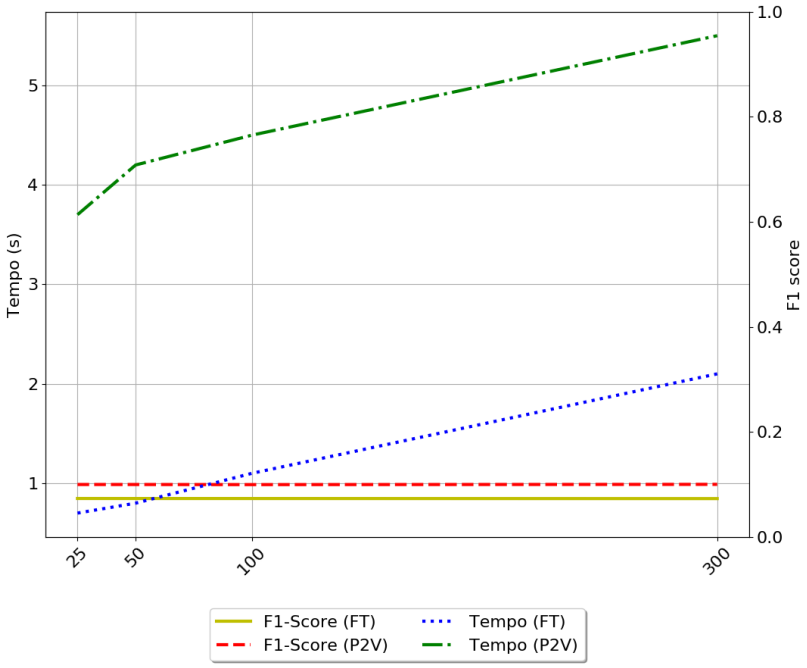
Fonte: Os Autores

de atualização de aprendizado (*learnDropRate*) é a medida da redução da taxa de aprendizado (*learnRate*) entre as iterações do algoritmo. Esse parâmetro assegura que novas amostras contribuam menos para um modelo que esteja convergindo para uma solução. Quanto menor o seu valor, menor é a contribuição de novas amostras. Ambos parâmetros contribuem pouco para a *F1-score* do algoritmo, como pode ser visto nas figuras 5.8 e 5.9. Além disso, eles não influenciam significativamente o custo computacional.

Função de custo ou função de perda (*lossFunction*) é a etapa final de um algoritmo de rede neural. Esse valor é uma métrica da qualidade do modelo, especificamente ele representa o custo a ser pago pela imprecisão das predições. O objetivo do treinamento é minimizar esse custo. Nos modelos desse trabalho, o valor da consiste no erro do mapeamento entre um documento ou palavra e seu contexto. Os modelos possuem três opções de função de custo: *softmax*, *hierarchical softmax* e *negative sampling*.

A função de custo *hierarchical softmax*, desenvolvida por Frederic Morin e Yoshua Bengio, consiste em aproximar *softmax* decompondo a tarefa de normalização. (MORIN; BENGIO, 2005) Em vez de executar a tarefa para todas as palavras do vocabulário, as palavras de contexto são organizada nas folhas de uma árvore binária e a normalização ocorre entre os nós irmãos a partir da raiz. A variante usada nos algoritmos derivados de *Word2Vec* leva em conta a frequência das palavras e usando uma árvore de Huffmann no lugar de uma binária.(MIKOLOV et al., 2013b) A última função, *negative sampling*,

Figura 5.6: Tamanho do Vetor



Fonte: Os Autores

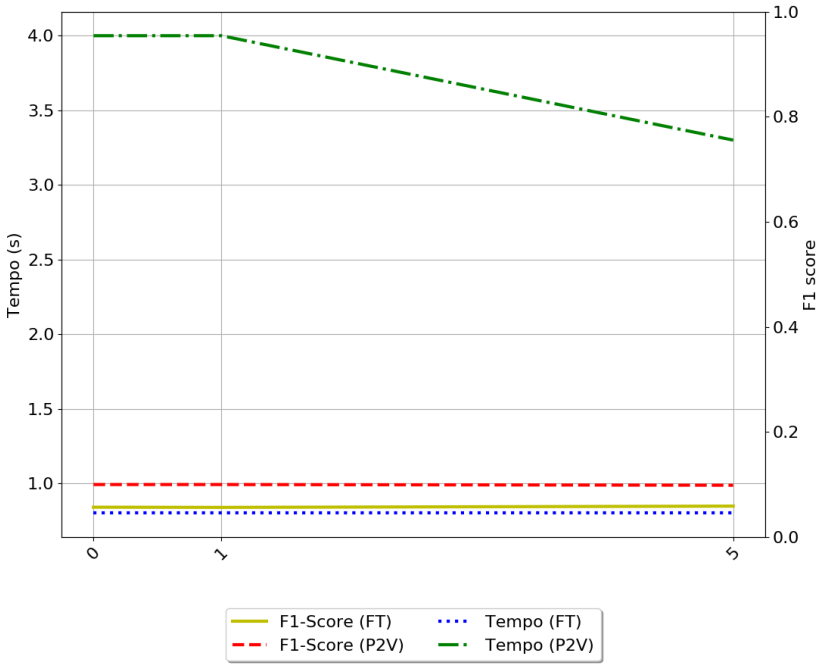
consiste aplicar a função *softmax* apenas para as palavras de contexto e uma amostra do restante do vocabulário.

Nas figuras 5.10 e 5.10, são analisadas as funções *negative sampling* (ns), com amostras de tamanho 5 e 1; *softmax* e *hierarchical softmax* (hs), respectivamente. Pode-se analisar nas figuras um comportamento contrastante: *Doc2Vec* apresenta os melhores resultados com *negative sampling*, enquanto *fastText* possui melhores resultados com *hierarchical softmax*. Além disso, a função de custo influencia o custo computacional do algoritmo *fastText* mais do que *Doc2Vec*, o último apresentando menor custo computacional exceto com *hierarchical softmax*.

O último parâmetro analisado é o limiar de amostra (*sample threshold*). Esse parâmetro define a taxa de sub-amostragem das palavras do vocabulário. Quanto maior o limiar de amostra, maior a chance de palavras muito comuns serem descartadas do contexto. O objetivo dessa abordagem evitar que palavras muito frequentes influenciem demais a representação distribuída sendo treinada. Conforme pode ser visto na figura 5.11, esse parâmetro influencia mais os resultados do algoritmo *Doc2Vec*, apresentando uma excelente taxa de *F1-score*, acima de 97% para um limiar maior.

Por fim, analisamos como a métrica *F1-score* da classificação é afetada pelo pré-processamento do texto. Foram analisadas todas as combinações das seguintes técnicas de pré-processamento: remoção de *stopwords*, *stemming* e agrupamento de classes de

Figura 5.7: Número Mínimo de Ocorrências



Fonte: Os Autores

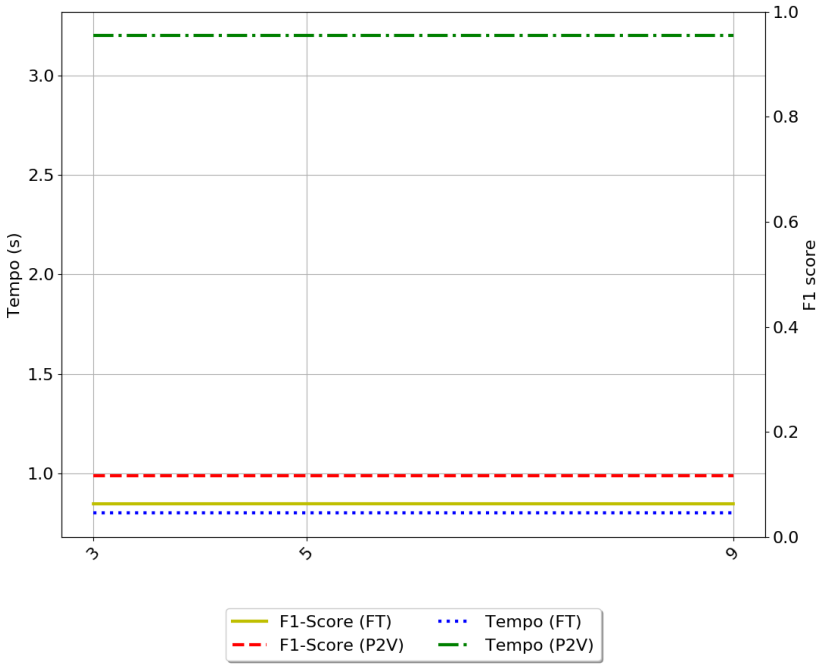
termos. Na figura 5.12, os pontos no eixo horizontal representam cada uma dessas opções dentro dos parênteses, segundo a ordem da legenda. Sendo o valor um correspondente a utilização da técnica e zero a não utilização.

Pode-se perceber que *fastText* é mais sensível ao pré-processamento do texto, obtendo resultados satisfatórios em apenas uma configuração. Apenas a remoção de *stopwords* mostra-se necessária para bons resultados. Já o modelo *Doc2vec* apresenta bons resultados sem remoção de *stopwords* e com a opção de *stemming* habilitada.

5.2 Avaliação dos Modelos

Ambos modelos apresentaram excelentes resultados para a precisão da classificação, com *Doc2Vec* alcançando mais de 97% de *F1-score* e *fastText* mais de 96%. A maior diferença na qualidade da classificação entre os modelos mostra-se na revocação, onde *fastText* alcança pouco mais de 75% enquanto *Doc2Vec* pode chegar a 98%. Esse resultado pode estar sendo influenciado pela metodologia de seleção dos *tweets* candidatos. Muitos dos *tweets* de trânsito pertenciam a contas especializadas em publicações ou referiam-se a elas. Dentre os erros de classificação, muitos incluem vocabulário pouco usado, por exemplo o seguinte *tweet* não é classificado como incidente de trânsito em

Figura 5.8: Tamanho da Janela de Contexto

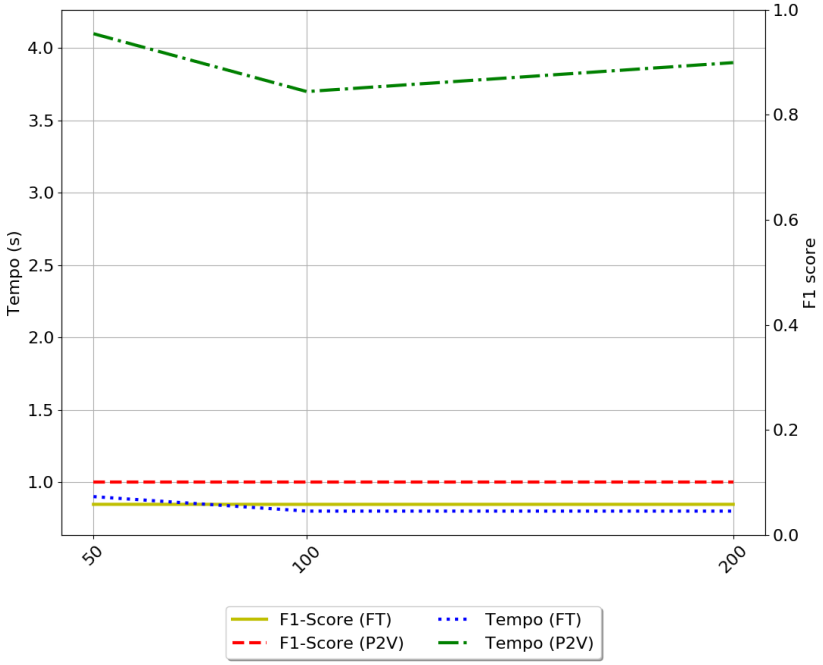


Fonte: Os Autores

algumas configurações dos algoritmos: "Içamento do vão móvel começa em 10 minutos @triunfoconcepa". Isso provavelmente ocorre, pois esse *tweet* possui uma estrutura muito diferente dos demais. Ele relata um evento que interferiu no trânsito após a data de sua publicação, em vez de um evento que esteja acontecendo ou já tenha acontecido. Ainda assim, o resultado é muito positivo, pois o conjunto de teste é muitas ordens de magnitude menor do que outros trabalhos.

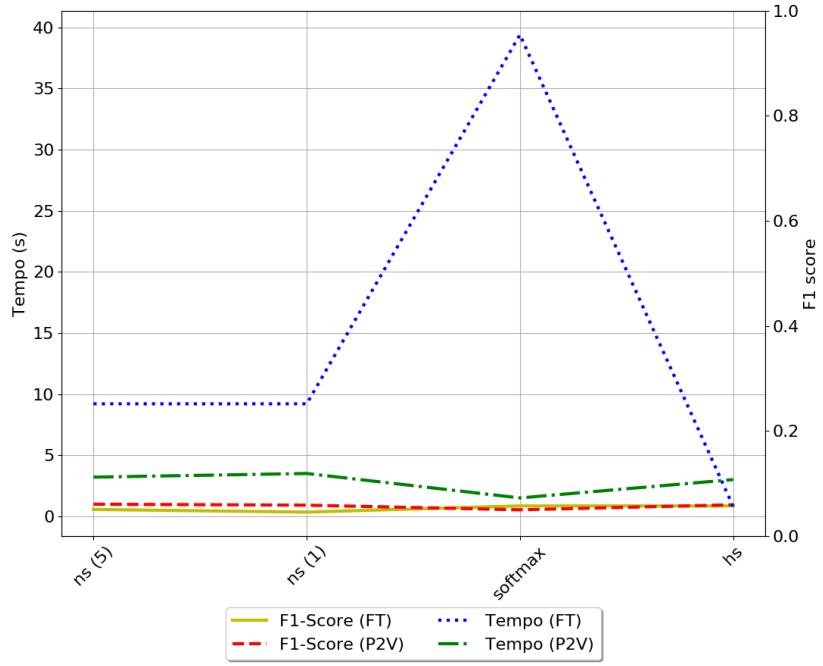
Outro resultado interessante é o baixo custo computacional dos classificadores. Nessa métrica *fastText* se destaca, sendo capaz de treinar seus modelos de maior qualidade em menos de um segundo. O modelo *Doc2Vec* também apresenta bons resultados na faixa de 3 a 5 segundos. Cabe salientar que os teste foram executados em um *notebook* com uma arquitetura de baixo consumo, conforme descrito na seção 4.1.

Figura 5.9: Taxa de Atualização de Aprendizado



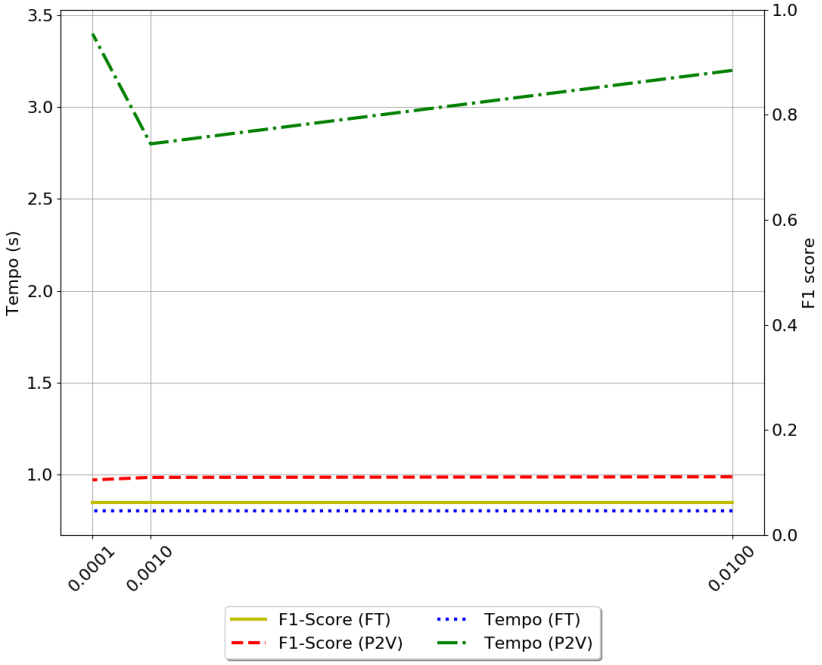
Fonte: Os Autores

Figura 5.10: Função de Custo



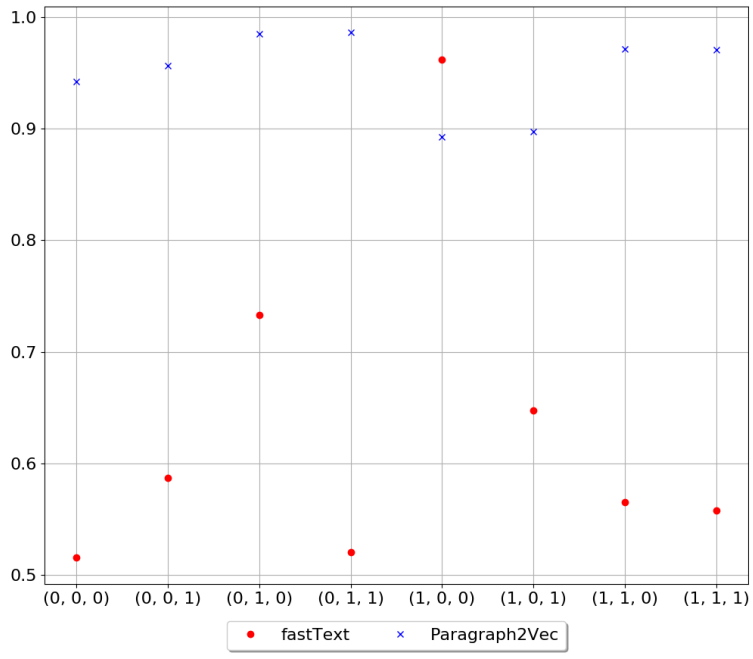
Fonte: Os Autores

Figura 5.11: Limiar de Amostra



Fonte: Os Autores

Figura 5.12: F1-score - Pré-processamento



Fonte: Os Autores

6 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Nesse trabalho foi um programa capaz de coletar *tweets* provenientes da cidade de Porto Alegre e identificar quais deles relatam incidentes de trânsito. Para esse fim foi implementado um classificador usando técnicas de *word embedding* e aprendizado supervisionado. Além disso, dois modelos de *word embedding* para documentos foram analisados por meio desse classificador: *Doc2Vec* e *fastText*. Ambos modelos são capazes de classificar corretamente os *tweets* na grande maioria dos casos e possuem um baixo custo computacional.

6.1 Dificuldades Encontradas

Uma das maiores dificuldades ao se trabalhar com questões de trânsito urbano é obtenção de informações com livre acesso, um dos diferenciais desse trabalho, é justamente utilizar o Twitter, por disponibilizar suas informações, dentro de certas limitações para usuários comuns, como:

1. Por usuário ou por aplicativo: é criado um *token* de acesso que tem um limite de 15 autenticações;

2. Janelas de 15 minutos: a cada 15 minutos pode ser solicitado um novo *token* de acesso para mais 15 autenticações.[1]

Os usuários *premium* não possuem essas limitações. Dessa forma, uma das primeiras dificuldades encontrada foi a limitação de acesso da API do Twtitter.

Por conta da escolha pelo aprendizado supervisionada, o trabalho exige rotulação manual dos documentos. Essa etapa demanda muito tempo para ser feita, uma vez que para os treinos e avaliação do classificador são necessários muitos *tweets*.

Grande parte das postagens feitas sobre trânsito na cidade de Porto Alegre são provenientes de páginas oficiais e específicas sobre o assunto, como @EPTC_POA, @TransitoPOARS, @transitogzh. Essas páginas, em sua maioria, produzem e compartilham mensagens padronizadas que possuem formatos semelhantes, como por exemplo:

(Horário da ocorrência) - informação de intercorrência de trânsito - em linguagem formal.

[1] https://developer.twitter.com/en/docs/basics/rate-limiting.html

Dessa forma, é preciso ter um cuidado redobrado para que o classificador não aprenda a diferenciação dos *tweets* de trânsito e de outros tópicos de forma tendenciosa. Contrastando a isso, também temos que a linguagem utilizada pela maioria do usuários-padrão da rede social, é informal, uma vez que são utilizados muitas gírias e abreviações que decorrem das limitações de caracteres do serviço.

As ferramentas utilizadas também apresentaram certa instabilidade durante a implementação do classificador. Por exemplo, a biblioteca *Gensim* possui uma implementação parcial das funcionalidades *fastText* até o momento da realização desse trabalho. O repositório de projetos *Python Pypi*[2] possui uma implementação instável da interface *Python* para *fastText*. A solução recorrida foi compilar as interfaces *Python* disponibilizadas no repositório oficial.[3] Essa implementação não suporta oficialmente sistemas operacionais *Windows*, que era o ambiente de desenvolvimento original desse trabalho, exigindo migração para outro sistema operacional.

6.2 Trabalhos Futuros

Os recursos na área de processamento de linguagem natural em português são escassos em relação à língua inglesa. Por isso, este trabalho possui muitas possibilidades de melhoria. A etapa de pré-processamento pode ser expandida para incluir técnicas como identificação de entidade nomeada, importante para análise de locais dos incidentes e lematização que pode apresentar resultados superiores ao *stemming*. Ambos modelos permitem incluir modelos pré-treinados de vetores de palavras. Com isso, pode-se aliar o modelo supervisionado aqui implementado com vetores treinados. As idiossincrasias do *Twitter*, gírias e erros de digitação, como podem ser tratadas explicitamente aprendendo *word embeddings* sobre um corpus maior.

Neste trabalho os termos de busca foram adquiridos apenas previamente ao aprendizado e a relevância de cada um deles não foi analisada. Por isso, seria interessante desenvolver um sistema de aprendizado dos termos de busca. Esse sistema poderia aprender novos termos conforme são coletados novos *tweets* e identificar sua relevância para filtrar incidentes de trânsito.

O foco desse trabalho foi o desenvolvimento e análise de um classificador de *tweets*. Contudo, os incidentes detectados pelos *tweets* classificados não são analisados. Uma

[2]https://pypi.org/

[3]https://github.com/facebookresearch/fastText/tree/master/python

possibilidade de trabalho futuro é a categorização dos incidentes, por exemplo, identificando se o *tweet* relata um congestionamento ou acidente. Além disso, pode-se coletar informações sobre extensão, horário, duração e local dos incidentes. Aliando-se os pontos anteriores, pode-se remodelar o sistema atual como um serviço, permitindo a visualização de incidentes de trânsito, integração com sistemas já existentes e a interação do usuário para melhorar sua qualidade.

REFERÊNCIAS

AFONSO, S. et al. Floresta sintá (c) tica: a treebank for portuguese. In: ELRA. **quot; In Manuel González Rodrigues; Carmen Paz Suarez Araujo (ed) Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002)(Las Palmas de Gran Canaria Espanha 29-31 de Maio de 2002) Paris: ELRA**. [S.l.], 2002.

AGRAWAL, A. W. et al. **Comparing Data Quality and Cost from Three Modes of On-Board Transit Passenger Surveys**. 2015.

BEER, D. Using social media data aggregators to do social research. **Sociological Research Online**, SAGE Publications Sage UK: London, England, v. 17, n. 3, p. 1–12, 2012.

BLOOM, D. E.; CANNING, D.; FINK, G. Urbanization and the wealth of nations. v. 319, 02 2008.

BOJANOWSKI, P. et al. Enriching word vectors with subword information. **arXiv preprint arXiv:1607.04606**, 2016.

BRAMBILLA, A. et al. Para entender as mídias sociais. **Creative commons**, 2011.

BRASIL. **Paramêtros Curriculares Nacionais: terceiro e quarto ciclos do ensino fundamental: língua portuguesa**. [S.l.]: MEC/SEF, 1998.

BROWN, P. F. et al. The mathematics of statistical machine translation: Parameter estimation. **Computational linguistics**, MIT Press, v. 19, n. 2, p. 263–311, 1993.

CEGALLA, D. P. **Novíssima Gramática da Língua Portuguesa - Novo Acordo Ortográfico**. [S.l.]: Companhia Editora Nacional, 2009.

CHEN, Y. et al. A convolutional neural network for traffic information sensing from social media text. In: IEEE. **Intelligent Transportation Systems (ITSC), 2017 IEEE 20th International Conference on**. [S.l.], 2017. p. 1–6.

CHOWDHURY, M.; SADEK, A. **Fundamentals of Intelligent Transportation Systems Planning**. Artech House, 2003. (Artech House ITS library). ISBN 9781580531603. Disponível em: <https://books.google.com.br/books?id=Zut3O7jVIiUC>.

COELHO, T. **10 fatos sobre o uso de redes sociais no Brasil que você precisa saber**. 2018. Disponível em: <https://www.techtudo.com.br/noticias/2018/02/10-fatos-sobre-o-uso-de-redes-sociais-no-brasil-que-voce-precisa-saber.ghtml>.

COSTE, R. G. D.; GALLISON, R. Dicionário de didáctica das línguas. **Coimbra, Editora Almedina**, 1983.

CUI, J. et al. Extraction of traffic information from social media interactions: Methods and experiments. In: IEEE. **Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on**. [S.l.], 2014. p. 1549–1554.

D'ANDREA, E. et al. Real-time detection of traffic from twitter stream analysis. **IEEE transactions on intelligent transportation systems**, IEEE, v. 16, n. 4, p. 2269–2283, 2015.

DARIDO, I. G. B. P. G. B. Planejamento em sistemas de transportes inteligentes (its) - perspectivas das experiências internacionais. **Série de Cadernos Técnicos - Sistemas Inteligentes de Transportes**, v. 8, p. 10–48, 2012.

DAVIES, P. The singularity is near: When humans transcend biology. **Nature**, v. 440, n. 7083, p. 421–422, 2006.

G1. **Mobilidade é o grande problema de Porto Alegre, aponta pesquisa**. 2013. Disponível em: <http://g1.globo.com/rs/rio-grande-do-sul/noticia/2013/09/mobilidade-e-o-grande-problema-de-porto-alegre-aponta-pesquisa.html>.

GAKENHEIMER, R. Urban mobility in the developing world. **Transportation Research Part A: Policy and Practice**, v. 33, n. 7, p. 671 – 689, 1999. ISSN 0965-8564. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0965856499000051>.

GAUCHAZH. **Por que as Ruas de Porto Alegre Engarrafam**. 2017. Disponível em: <https://gauchazh.clicrbs.com.br/transito/noticia/2017/06/por-que-as-ruas-e-avenidas-de-porto-alegre-engarrafam-9823677.html>.

GOLDBERG, Y. Neural network methods for natural language processing. **Synthesis Lectures on Human Language Technologies**, Morgan & Claypool Publishers, v. 10, n. 1, p. 1–309, 2017.

GU, Y.; QIAN, Z. S.; CHEN, F. From twitter to detector: Real-time traffic incident detection using social media data. **Transportation research part C: emerging technologies**, Elsevier, v. 67, p. 321–342, 2016.

HARRIS, Z. S. Distributional structure. **Word**, Taylor & Francis, v. 10, n. 2-3, p. 146–162, 1954.

HAYDUK, B. et al. **Multimodal Transportation Planning Data: Compendium of Data Collection Practices and Sources**. Transportation Research Board (TRB), 1997. (NCHRP Web document). Disponível em: <https://books.google.com.br/books?id=9cR2AQAACAAJ>.

HUANG, Z. **Data integration for urban transport planning**. Tese (Doutorado) — Utrecht University, 2003.

JURAFSKY, D.; MARTIN, J. H. **Speech and language processing**. [S.l.]: Pearson London:, 2014. v. 3.

KAPPEL, I. B. A. **Segmentação textual, coesão e distribuição informacional na organização tópica do texto**. Tese (Doutorado) — Dissertação de mestrado. Uberlândia, UFU/ILEEL: 165 p, 1998.

LE, Q.; MIKOLOV, T. Distributed representations of sentences and documents. In: **International Conference on Machine Learning**. [S.l.: s.n.], 2014. p. 1188–1196.

LEAL, A. G.; SANTOS, A. S. D. Tendências e caminhos das pesquisas em sistemas inteligentes de transporte. 2015.

LUCCA, J. D.; NUNES, M. d. G. V. Lematização versus stemming. **USP, UFSCar, UNESP, São Carlos, São Paulo**, 2002.

MARCUS, M. P.; MARCINKIEWICZ, M. A.; SANTORINI, B. Building a large annotated corpus of english: The penn treebank. **Computational linguistics**, MIT Press, v. 19, n. 2, p. 313–330, 1993.

MARQUES, L. K. d. S.; VIDIGAL, F. Prosumers and social networks as marketing information sources. an analysis from the perspective of competitive intelligence in brazilian companies. **Transinformação**, SciELO Brasil, v. 30, n. 1, p. 1–14, 2018.

MARTINS, S. R. et al. Geração automática de textos em plataformas de governo eletrônico: um estudo de caso na plataforma lattes. Florianópolis, SC, 2004.

MCQUEEN, B.; MCQUEEN, J. **Intelligent transportation systems architectures**. [S.l.: s.n.], 1999.

MIKOLOV, T. et al. Efficient estimation of word representations in vector space. **arXiv preprint arXiv:1301.3781**, 2013.

MIKOLOV, T. et al. Distributed representations of words and phrases and their compositionality. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2013. p. 3111–3119.

MIKOLOV, T.; YIH, W.-t.; ZWEIG, G. Linguistic regularities in continuous space word representations. In: **Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**. [S.l.: s.n.], 2013. p. 746–751.

MILNE, D.; WATLING, D. Big data and understanding change in the context of planning transport systems. **Journal of Transport Geography**, 2018. ISSN 0966-6923. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0966692317300984>.

MORIN, F.; BENGIO, Y. Hierarchical probabilistic neural network language model. In: CITESEER. **Aistats**. [S.l.], 2005. v. 5, p. 246–252.

NASCIMENTO, P. et al. Análise de sentimento de tweets com foco em notícias. 07 2012.

NATIONS, U. **World Urbanization Prospects: The 2018 Revision, Online Edition**. 2014. Disponível em: <https://esa.un.org/unpd/wup/>.

ORENGO, V.; HUYCK, C. A stemming algorithmm for the portuguese language. In: IEEE. **spire**. [S.l.], 2001. p. 0186.

PERINI, M. A. et al. Sobre a Classificação das Palavras. **DELTA: Documentação de Estudos em Linguística Teórica e Aplicada**, scielo, v. 14, p. 00 – 00, 00 1998. ISSN 0102-4450. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0102-44501998000300014&nrm=iso>.

ŘEHŮŘEK, R.; SOJKA, P. Software Framework for Topic Modelling with Large Corpora. In: **Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks**. Valletta, Malta: ELRA, 2010. p. 45–50. <http://is.muni.cz/publication/884893/en>.

SAKAKI, T. et al. Real-time event extraction for driving information from social sensors. In: IEEE. **Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2012 IEEE International Conference on**. [S.l.], 2012. p. 221–226.

SANTOS, M. H. P. Em torno dos conceitos operatórios de 'sintaxe', de 'frase', de 'oração'e de 'enunciado'utilizados numa gramática novecentista da língua portuguesa. **Acta Scientiarum. Language and Culture**, Universidade Estadual de Maringá, v. 36, n. 4, 2014.

SCHULZ, A.; RISTOSKI, P.; PAULHEIM, H. I see a car crash: Real-time detection of small scale incidents in microblogs. In: SPRINGER. **Extended Semantic Web Conference**. [S.l.], 2013. p. 22–33.

UYSAL, A. K.; GUNAL, S. The impact of preprocessing on text classification. **Information Processing & Management**, Elsevier, v. 50, n. 1, p. 104–112, 2014.

VASCONCELLOS, E. de. **Urban Transport, Environment, and Equity: The Case for Developing Countries**. Earthscan Publications, 2001. ISBN 9781853837272. Disponível em: <https://books.google.com.br/books?id=MyBPAAAAMAAJ>.

VITRAL, L. O que é gramatica gerativa? **Anais da Semana de Estudos de Língua Portuguesa**, v. 2, n. 1, p. 69–72, 1996.

WANG, S. et al. Traffic condition analysis based on users emotion tendency of microblog. In: SPRINGER. **UK Workshop on Computational Intelligence**. [S.l.], 2017. p. 299–311.

WANICHAYAPONG, N. et al. Social-based traffic information extraction and classification. In: IEEE. **ITS Telecommunications (ITST), 2011 11th International Conference on**. [S.l.], 2011. p. 107–112.

WHITING, A.; WILLIAMS, D. Why people use social media: a uses and gratifications approach. **Qualitative Market Research: An International Journal**, Emerald Group Publishing Limited, v. 16, n. 4, p. 362–369, 2013.

WILLIAMS, B. **Intelligent transport systems standards**. [S.l.]: Artech House, 2008.

WILSON, R. A.; KEIL, F. C. **The MIT encyclopedia of the cognitive sciences**. [S.l.]: MIT press, 2001.