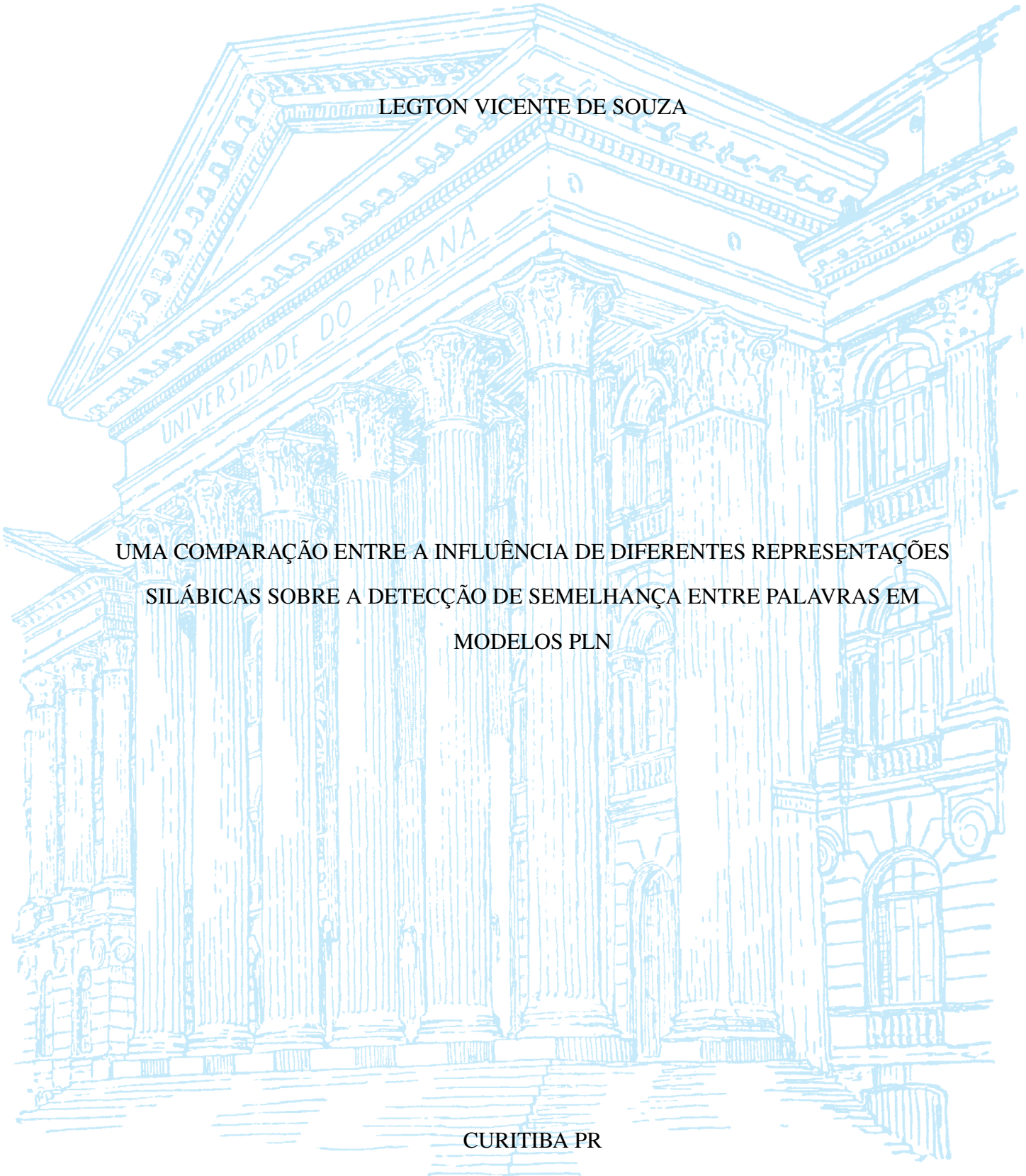
UNIVERSIDADE FEDERAL DO PARANÁ

LEGTON VICENTE DE SOUZA



UMA COMPARAÇÃO ENTRE A INFLUÊNCIA DE DIFERENTES REPRESENTAÇÕES SILÁBICAS SOBRE A DETECÇÃO DE SEMELHANÇA ENTRE PALAVRAS EM MODELOS PLN

CURITIBA PR

2022

LEGTON VICENTE DE SOUZA

UMA COMPARAÇÃO ENTRE A INFLUÊNCIA DE DIFERENTES REPRESENTAÇÕES SILÁBICAS SOBRE A DETECÇÃO DE SEMELHANÇA ENTRE PALAVRAS EM MODELOS PLN

Trabalho apresentado como requisito parcial à conclusão do Curso de Bacharelado em Ciência da Computação, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Fabiano Silva.

CURITIBA PR

2022

RESUMO

O texto escrito é uma das formas mais básicas de comunicação, com uma abundância de conteúdo disponível em diferentes mídias. Algoritmos de processamento de linguagens naturais (PLN) constituem diferentes técnicas para a análise desse conteúdo, construindo modelos computacionais capazes de capturar a complexidade de uma linguagem. Apesar disso, certas características das palavras do vocabulário de um idioma não são facilmente perceptíveis, gerando uma complexidade maior no aprendizado por estes algoritmos. Utilizando algoritmos de PLN, mais especificamente os algoritmos baseados em *word embedding* Word2Vec e FastText, este trabalho se propõe a avaliar o impacto de uma representação alternativa das palavras de um vocabulário na detecção de semelhanças entre as palavras deste, gerando agrupamentos de palavras semelhantes levando em consideração características ortográficas e fonéticas de cada palavra. Para comparação da eficácia de tal representação, foram treinados modelos Word2Vec e FastText para três diferentes representações de palavras: divisão silábica, fonética e uma nova técnica de etiquetagem proposta durante o desenvolvimento do projeto Scriba, com os modelos sendo comparados sobre a detecção de palavras similares para um conjunto de palavras pré-definido. A categorização pela etiquetagem proposta no projeto Scriba foi capaz de identificar semelhanças entre palavras que não foram detectadas pela representação usual ou fonética de palavras.

Palavras-chave: Processamento de Linguagem Natural. Word2Vec. FastText.

CONTENTS

1 INTRODUÇÃO

O texto escrito é uma das formas mais básicas de comunicação, estando omnipresente nos diferentes tipos de mídia. Desta forma, há uma abundância de informação textual existente disponível para análise através de técnicas computacionais. Apesar disso, existe certa dificuldade em replicar mecanismos de leitura humana sobre esses textos para identificar similaridades, características ou padrões que podem ser observados através da análise desses textos. O objetivo da área de processamento de linguagens naturais (PLN) é estudar algoritmos e métodos para a construção de modelos computacionais que sejam capazes de analisar linguagens naturais e capturar suas complexidades (Ly et al., 2020; Khan et al., 2016). Em geral, aplicações de algoritmos de processamento de linguagens naturais tem como propósito facilitar a comunicação humana, auxiliando em dificuldades linguísticas, e também melhorar a comunicação humano-computador, facilitando o processamento de informação.

Um caso de uso específico no qual o processamento de linguagens naturais pode ser aplicado é para auxiliar no desenvolvimento da educação de alunos de ensino fundamental. O processo de aprendizagem de regras ortográficas é um processo lento e complexo, e para alunos com dificuldades de aprendizado, assimilar as regras de estrutura ortográfica é um processo difícil, com erros persistindo em diversos graus de educação (Zanella, 2011). A aplicação de algoritmos de PLN pode auxiliar na identificação de padrões nas dificuldades apresentadas por diferentes grupos de alunos com o objetivo de planejar o ensino de forma mais adequada a facilitar o aprendizado. O objetivo deste trabalho é - através da utilização de algoritmos de processamento de linguagens naturais - validar a proposta de uma estratégia de etiquetagem de palavras em uma nova estrutura genérica para a representação ortográfica que considera características fonéticas silábicas. Esta estratégia está sendo proposta durante o desenvolvimento do projeto de pós-doutorado Scriba (Adelaide H. P. Silva, 2022), uma inteligência artificial aplicada ao ensino de ortografia. Neste trabalho, este processo de validação se dá pela comparação entre os resultados obtidos em detecção de palavras similares e padrões analisados por algoritmos de PLN. Diferentes representações de palavras são utilizadas, com o objetivo de detectar se a análise de textos com a etiquetagem proposta pode apresentar a detecção de padrões não usualmente perceptíveis pelas representações silábicas e fonéticas de um vocabulário, possivelmente auxiliando na detecção de erros de palavras que podem ser agrupadas em categorias que auxiliem no ensino de ortografia.

Para realizar os experimentos, dois métodos de construção de modelos de linguagens naturais baseados em *word embedding* foram utilizados: Word2Vec (Mikolov et al., 2013) e FastText (Bojanowski et al., 2017). Diferentes modelos foram treinados utilizando diferentes representações de palavras extraídas do dataset Aeiouadô (Mendonça and Aluísio, 2014). Resultados positivos foram encontrados ao utilizar a etiquetagem proposta em modelos treinados utilizando o Word2Vec. Apesar disso, este método possui o ponto negativo de que modelos gerados utilizando-o não conseguem detectar similaridades para com palavras fora do vocabulário aprendido pelo modelo. Para isso, modelos utilizando o método FastText também foram treinados. Apesar de conseguir identificar certo nível de similaridade mesmo para palavras fora do vocabulário, estes modelos apresentaram menor nível de similaridade com palavras baseando-se na etiquetagem proposta, dadas as características desse método de focar em similaridades entre conjuntos de caracteres das palavras do vocabulário.

O restante do trabalho está organizado da seguinte forma. O Capítulo 2 apresenta conceitos básicos da representação ortográfica e fonética da linguagem portuguesa. Em seguida, é apresentada uma nova representação de estrutura ortográfica genérica em desenvolvimento.

No Capítulo 3, são apresentados conceitos básicos do processamento de linguagens naturais, assim como dois métodos de aprendizado de *word embeddings* para a construção dos modelos apresentados nos experimentos realizados: Word2Vec e FastText. Por fim, o Capítulo 4 apresenta os modelos treinados e descreve os resultados obtidos ao comparar os diferentes modelos. As conclusões seguem no Capítulo 5.

2 CONCEITOS BÁSICOS

No contexto de processamento de linguagens naturais na lingua portuguesa, é importante conhecermos alguns conceitos básicos sobre como palavras podem ser representadas, levando em consideração aspectos como sílabas tônicas e fonética. Neste capítulo, apresentamos inicialmente a representação por separação silábica indicando silabas tônicas e características de fonéticas como dadas pelo alfabeto fonético internacional. Em seguida, apresentamos a etiquetagem sendo proposta no trabalho de pós-doutorado Scriba e explicamos seu funcionamento.

2.1 SEPARAÇÃO SILÁBICA E REPRESENTAÇÃO FONÉTICA

Toda palavra pode ser separada em sílabas, com diversas regras ortográficas se aplicando na divisão. Apesar disso, em certos casos é importante denotar a silaba tônica de uma palavra, visto que isto não é visível em certos casos pela falta de um acento. Para os propósitos desse trabalho, a sílaba tônica é representada por um apóstrofo (') no inicio da sílaba. Por exemplo, para a palavra "açúcar", temos a seguinte representação:

a.'çú.car

Uma outra possível representação para palavras é a utilização do alfabeto fonético internacional (Association et al., 1999). Desta forma, ao invés de representar as sílabas utilizando o alfabeto comum da lingua portuguesa, podemos utilizar a representação fonética das silabas. Isso é particularmente útil porque, em certos casos, consoantes podem gerar o mesmo som (como é o caso com consoantes z e s em certas palavras), permitindo uma classificação mais precisa de certas palavras com base em sua fonética. Para a palavra "açúcar", podemos representar a divisão silábica fonética da seguinte forma, levando em consideração também o indicador da sílaba tônica:

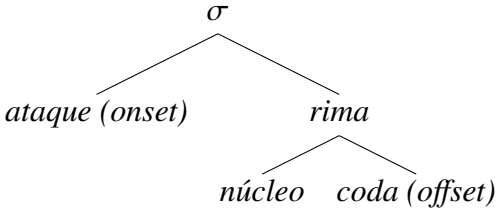
a.'su.kəx

2.2 A ETIQUETAGEM PROPOSTA

Uma representação alternativa para as palavras de um vocabulário proposta por (Adelaide H. P. Silva, 2022) é validada neste trabalho através do treinamento dos modelos e comparação com os resultados obtidos dos modelos treinados com as representações apresentadas anteriormente. Essa etiquetagem das palavras considera a estrutura silábica e acentual, descrevendo certos aspectos fonéticos silábicos de uma palavra. A Tabela 2.1 apresenta as etiquetas utilizadas para cada uma das variáveis e representações ortográficas das sílabas de uma palavra.

A unidade linguística escolhida para essa representação é a sílaba, visto que nesta podemos verificar o relacionamento entre sons da fala e também agrupar certas características desses sons.

A Linguística considera a seguinte estrutura interna para sílabas:



Variável	Representação Ortográfica	Etiqueta
Consoantes oclusivas	p, b, t, d, c, qu, g, gu	O
Consoantes fricativas	f, v, s, ss, c, x, z, ch, j, g	F
Consoantes nasais	m, n, nh	N
Consoantes líquidas	l, lh, r, rr	L
Vogais	i, e, a, o, u, ã, õ	V
Ataque simples	O, F, N ou L	SA
Núcleo simples	V	SN
Coda simples	p, t, d, c, g, f, s, z, m, n, l, r	SC
Primeira posição de ataque ramificado	p, b, t, d, c, g, f, v	CA1
Segunda posição de ataque ramificado	l, r, s, m, n	CA2
Primeira posição de núcleo complexo	i, e, a, o, u, ã, õ	CN1
Segunda posição de núcleo complexo	i, u, e, o	CN2
Primeira posição de coda complexa	n, r	CC1
Segunda posição de coda complexa	s	CC2
Sílaba tônica		3
Sílabas pretônica e postônica		1
Sílaba postônica átona final		0

Table 2.1: Tabela de etiquetas. Fonte: Adelaide H. P. Silva (2022)

Nesta estrutura, os constituintes "ataque" e "coda" são opcionais e representam consoantes. O constituinte núcleo é obrigatório, e sempre representa uma vogal. Conforme a Tabela 2.1, esses constituintes são representados por "SA", "SC" e "SN" respectivamente. Cada um destes constituintes pode ser designado como "ramificado" ou "complexo", que indica a presença de mais de uma unidade sonora no constituinte silábico. Em uma coda complexa, temos o final da sílaba com duas consoantes, como em "**pers**pectiva". Para ataques complexos, temos um início de sílaba com duas consoantes, como em "**pr**aga". Para núcleos complexos, existe um encontro de duas vogais, como em "**cai**xa". A posição da unidade sonora nestes casos é marcada pelos índices 1 e 2.

As sílabas tônicas também influenciam na representação das etiquetas. Como disposto na Tabela 2.1, a sílaba tônica, sendo a sílaba mais intensa, recebe o maior grau de acento (3), enquanto sílabas pretônicas recebem o grau 1 e sílabas postônicas recebem o grau 0. O índice 2 foi reservado para o que é conhecido como "acento secundário" na literatura, não sendo contemplado neste momento na representação.

O restante das etiquetas (O, F, N, L, V) representam classes de sons, que desempenham papel importante como causas de erros ortográficos. Essas etiquetas são importantes visto que certas consoantes produzem sons similares, e modelos de processamento de linguagens naturais treinados sobre a representação das etiquetas podem ser capazes de perceber erros gramaticais de trocas de consoantes baseadas em suas categorias, como por exemplo uma oclusiva sendo trocada por outra oclusiva.

Ademais, na representação das etiquetas, cada som específico interno à silaba é agrupado em parênteses, e as sílabas são agrupadas em colchetes, com o grau de acento sendo informado em seguida. Um exemplo de representação de uma palavra neste sistema de etiquetagem pode ser dada para a palavra "açúcar". Conforme a estrutura apresentada, a etiquetagem desta palavra pode ser dada por [(SN)]1[(SAF)(SN)]3[(SAO)(SN)(SCL)]0. Para a primeira sílaba, (SN) corresponde ao núcleo da sílaba. Como nenhuma consoante está presente nesta sílaba, não possuímos constituintes de coda ou ataque. O grau de acento 1 é informado em seguida, visto

que a sílaba é pretônica. Para a segunda sílaba, (SAF) corresponde à consoante fricativa (F) do ataque silábico (SA), e (SN) corresponde novamente ao núcleo da sílaba. O grau de acento neste caso é 3, visto que esta é a silaba tônica da palavra. Por fim, temos a última sílaba, com (SAO) representando a consoante oclusiva (O) do ataque silábico (SA), (SN) como núcleo da sílaba e (SCL) representando uma consoante líquida (L) da coda silábica (SC). O grau de acento nesse caso é 0, visto que a sílaba é postônica.

A importância dessa etiquetagem na representação de uma palavra se dá pelo fato de que ela consegue categorizar sílabas de uma forma em que categorias de sons e estrutura ortográfica são levadas em consideração no agrupamento de palavras ao treinar modelos. Essas características não são facilmente identificadas na representação no alfabeto comum ou no alfabeto fonético internacional, mesmo que certos sons produzidos pela silaba sejam similares. Essa abordagem possivelmente permite a identificação de erros de mesma categoria, como a substituição de certas consoantes que produzem sons similares. Além disso, garante também as restrições fonotáticas da lingua, ou seja, as sequencias de sons permitidas em uma lingua, visto que não permite a representação de sequências de letras como <shr>, que não são permitidas no português brasileiro.

3 MODELOS BASEADOS EM WORD EMBEDDING

Para realizar a comparação entre os agrupamentos das palavras do vocabulário utilizado nas diferentes representações, o primeiro passo é a construção de modelos a serem treinados para detectar tais similaridades. Este capítulo inicia apresentando a definição e funcionamento do conceito de *word embeddings* no contexto do processamento de linguagens naturais. Em seguida, apresenta a técnica de PLN conhecida como Word2Vec (Mikolov et al., 2013), descrevendo seu funcionamento básico, vantagens e desvantagens. As mesmas características são discutidas em seguida para o método FastText (Bojanowski et al., 2017).

3.1 WORD EMBEDDING

No contexto de processamento de linguagens naturais, o conceito de *word embedding* consiste na representação de palavras na forma de um vetor de valores reais, de tal forma que palavras próximas no espaço vetorial sejam similares. Cada palavra é mapeada para um vetor, usualmente de dimensões na casa de dezenas ou centenas. Cada dimensão do vetor de uma palavra representa uma característica detectada para aquela palavra durante o treinamento do modelo com base no conjunto de dados alimentado para o treinamento. É importante notar que essas características detectadas não são definidas manualmente e não necessariamente tem significado claro, sendo aprendidas pelo modelo durante o treinamento com base no contexto em que diferentes palavras estão sendo utilizadas e a similaridade do contexto com outras palavras.

Como as representações vetoriais de cada uma das palavras do vocabulário é influenciada fortemente pelo contexto no qual elas são utilizadas, palavras utilizadas em contextos similares possuem representações similares, permitindo que agrupamentos sejam realizados e que palavras similares sejam facilmente detectadas. Além disso, é importante observar que o *corpus* (o conjunto de dados de um texto alimentado ao modelo) não necessariamente precisa ser constituído por frases corretas semanticamente. Isto é, o *corpus* pode ser um conjunto de "frases" compostas por palavras que possuem algum tipo de relação, como certas características ortográficas em comum.

3.2 WORD2VEC

O algoritmo Word2Vec (Mikolov et al., 2013) baseia-se na utilização do conceito de *word embedding*, ou seja, na representação das palavras de um *corpus* como vetores em um espaço real. Esse algoritmo utiliza um modelo de rede neural para aprender as relações entre palavras a partir do *corpus* fornecido, permitindo detectar palavras "sinônimas" (mais especificamente, palavras com utilizadas em contextos similares). Além disso, o modelo treinado também permite a sugestão de possíveis palavras a serem utilizadas no contexto de uma frase parcial. O Word2Vec possui duas arquiteturas diferentes: *continuous bag of words* (CBOW) e *skip-gram*.

A arquitetura CBOW tenta prever palavras alvo com base no contexto em que elas se encontram, ou seja, as palavras ao redor da palavra alvo. Considere o pangrama "À noite, vovô Kowalsky vê o ímã cair no pé do pinguim queixoso e vovó põe açúcar no chá de tâmaras do jabuti feliz.". São construídos pares *(janela, palavra_alvo)* para cada palavra, que influenciam na representação vetorial da mesma. Para uma janela de tamanho 2, possíveis pares a serem gerados para o pangrama seriam ([À, vovô], noite) e ([vovô, vê], Kowalsky), dentre outros. Modelos CBOW são muito mais rápidos de serem treinados quando comparados a *skip-gram* e são mais

precisos em seus resultados para palavras que acontecem com frequência alta. Apesar disso, o modelo não consegue boas representações para palavras mais raras do vocabulário.

A arquitetura de modelo *skip-gram* é, de forma genérica, o oposto de CBOW. Nesta arquitetura, dada uma palavra alvo, o que tentamos prever são as palavras utilizadas naquele mesmo contexto, ou seja, palavras ao redor da palavra alvo, com os pares gerados sendo compostos pela palavra alvo e cada um dos vizinhos dentro da janela. Ou seja, para uma janela de tamanho 2, teríamos pares como (vovô, a), (vovô, noite), (vovô, Kowalsky) e (vovô, vê). Quando comparado a modelos CBOW, modelos *skip-gram* conseguem uma representação mais precisa para conjuntos de dados de treino de tamanhos menores, conseguindo obter resultados bons até mesmo para palavras que ocorrem com menor frequência. Apesar disso, dadas suas características, o treinamento de tais modelos é mais lento que o treinamento para modelos CBOW.

Em geral, modelos Word2Vec conseguem obter boas representações para *corpus* de diferentes tamanhos. Apesar disso, a principal desvantagem do modelo Word2Vec é que este não consegue gerar uma representação para palavras fora do vocabulário obtido do *corpus* utilizado para treinar o modelo. Isso dificulta a utilização do modelo para certas possíveis aplicações de modelos de processamento de linguagens naturais, como a detecção de erros gramaticais, visto que o modelo não reconhecerá palavras não existentes no vocabulário.

3.3 FASTTEXT

Similarmente ao Word2Vec, o algoritmo FastText possui as mesmas duas arquiteturas (CBOW e *skip-gram*) disponíveis para treinar os modelos, além de também utilizar características internas da palavra (sub-palavras), permitindo até mesmo a utilização de palavras fora do vocabulário.

Nesse modelo, além da aplicação das arquiteturas CBOW e *skip-gram*, cada palavra também é considerada como sendo composta por *n-grams*, considerando características dessas sub-palavras compostas por cada *n-gram* na construção da representação das palavras no espaço vetorial. Por exemplo, para a palavra 'Brasil' e $n = 3$, temos ['<br', 'bra', 'ras', 'asi', 'sil', 'il>'] como a representação *n-gram* desta palavra. O vetor da representação da palavra 'Brasil' é assumido como a soma da representação de vetores de cada um dos elementos da representação *n-gram*.

Essa representação utilizando características baseadas em conjuntos de carecteres da palavra original permite que uma representação possa ser estimada para qualquer palavra, permitindo que palavras fora do vocabulário também sejam consideradas pelo modelo quando este é aplicado. Apesar disso, essa característica pode ser um ponto negativo dependendo de como os dados a serem treinados estão representados, visto que nem sempre os *n-grams* fornecem informações que fazem sentido para treinar a similaridade das palavras, especialmente em casos em que as palavras já foram tokenizadas, visto que o algoritmo não possui conhecimento sobre os diferentes tokens que podem ser utilizados.

4 EXPERIMENTOS E RESULTADOS

Neste trabalho, a proposta é treinar modelos com diferentes representações de palavras do vocabulário em português disponibilizado no dataset Aeiouadô (Mendonça and Aluísio, 2014), comparando os agrupamentos obtidos e quais padrões foram identificados pelos modelos treinados utilizando as diferentes representações. O objetivo de tal comparação é validar a proposta da nova estrutura genérica para a representação ortográfica que considera características fonéticas silábicas e verificar se modelos treinados utilizando essa nova representação levam a detecção de padrões diferentes dos obtidos pelas representações usuais. Este capítulo inicia descrevendo os modelos treinados e em seguida apresenta alguns dos resultados obtidos pela utilização do modelo.

4.1 MODELOS TREINADOS

Para treinar os modelos, o algoritmo Word2Vec necessita de um corpo de texto grande que permita que o processo de treinamento do modelo detecte os contextos nos quais as palavras do vocabulário ocorrem para que este crie as representações vetoriais com base nessas características de contexto. Neste trabalho, utilizamos o dataset Aeiouadô (Mendonça and Aluísio, 2014), que disponibiliza um vocabulário de palavras da lingua portuguesa representadas no alfabeto normal e no alfabeto fonético internacional (IPA). Como o propósito deste trabalho é encontrar similaridades entre as palavras em um nível de representação estrutural ortográfica e fonética, o nosso corpo de texto utilizado para treinar um modelo não é um conjunto de frases, mas sim um conjunto de representações das palavras do vocabulário. A unidade linguística aqui utilizada é a sílaba, e portanto, para os propósitos deste trabalho, o "contexto" em que uma palavra se encontra é a sua representação silábica de diferentes formas, conforme as representações apresentadas no Capítulo 2. Isso significa que cada palavra do nosso modelo tem sua representação vetorial sendo influenciada não por um contexto semântico, mas sim por suas características ortográficas e fonéticas, permitindo o agrupamento das palavras com base nestas características.

O objetivo do trabalho é realizar experimentos com modelos treinados a partir das três representações ortográficas e fonéticas apresentadas no Capítulo 2 para validar a representação baseada na etiquetagem. Desta forma, precisamos do mesmo corpo de "texto" (nosso conjunto de representações de cada palavra) representado de diferentes formas. Considere a palavra 'abacaxi'. As seguintes representações podem ser dadas:

- **Representacão silábica.** Exemplo: a.ba.ca.'xi
- **Representação fonética (IPA).** Exemplo: a.ba.ka.'ʃi
- **Nova etiquetagem proposta.** Exemplo: [(SN V)]1.[(SA O)(SN V)]1.[(SA O)(SN V)]1.[(SA F)(SN V)]3

Considerando estas representações, chegamos em quatro estratégias diferentes para gerar os corpos de texto a serem utilizados pelo algoritmo Word2Vec para realizar o treinamento e construção do modelo:

- **Frases simples** - cada "frase" do corpo de texto é dada por uma palavra original do vocabulário e uma das representações acima dividida em silabas. Exemplo: ["abacaxi", "a", "ba", "ca", "'xi"]

- **Frases com contador relativo a silaba** - cada “frase” do corpo de texto é dada por uma palavra original do vocabulário e uma das representações acima dividida em silabas, com silabas duplicadas sendo diferenciadas por um contador utilizado como prefixo. Essa estratégia garante que o modelo não seja influenciado no aprendizado a acreditar que palavras que possuem silabas similares duplicadas possuem um grau de similaridade maior. Exemplo: ["anarcocomunismo", "0-a", "0-nar", "0-co", "1-co", "0-mu", "0-’nis", "0-mo"]

- **Frases com “sub-palavras”** - cada “frase” do corpo de texto é dada por uma palavra original do vocabulário e uma das representações acima dividida em silabas. Além disso, são geradas todas sub-palavras possíveis com as silabas levando em consideração a ordem das mesmas na palavra original. Essa estratégia tem como objetivo tentar influenciar o aprendizado a detectar similaridade de contexto com base em sub-palavras que podem ocorrer dentro de palavras maiores, similarmente a estrategia do FastText mas considerando silabas ao invés de *n-grams*. Exemplo: ["abacaxi", "a", "ba", "ca", "’xi", "aba", "baca", "ca’xi", "abaca", "baca’xi", "abaca’xi"]

- **Frases com “sub-palavras” e contador** - cada “frase” do corpo de texto é dada por uma palavra original do vocabulário e uma das representações acima dividida em silabas, com silabas duplicadas sendo diferenciadas por um contador utilizado como prefixo. Além disso, são geradas todas sub-palavras possíveis com as silabas levando em consideração a ordem das mesmas na palavra original. As sub-palavras também possuem um prefixo de contador de número de ocorrências. O objetivo é combinar as duas estratégias anteriores e obter as vantagens de ambas estratégias. Exemplo: ["abacaxi", "0-a", "0-ba", "0-ca", "0-’xi", "0_aba", "0_baca", "0_ca’xi", "0_abaca", "0_baca’xi", "0_abaca’xi"]

Após treinar os modelos utilizando esses corpos de textos gerados conforme as estratégias acima, o vocabulário do modelo é forçadamente restrito somente ao conjunto de palavras do vocabulário original, dispensando as representações intermediárias utilizadas para o contexto das palavras enquanto mantém sua influência na representação dos vetores. Dessa forma, palavras que estavam no mesmo contexto (nesse caso, palavras que possuem semelhança em sua representação) antes do vocabulário ser restrito ainda são indicadas como palavras similares.

Além dos modelos Word2Vec, também foram treinados alguns modelos FastText utilizando a última estratégia citada para gerar o corpo de texto de treinamento para cada uma das representações ortográficas/fonéticas das palavras. Todos os modelos foram treinados utilizando a implementação dos algoritmos Word2Vec e FastText disponível pela biblioteca *gensim* (Řehřek et al., 2011) na linguagem Python.

4.2 RESULTADOS

Cada um dos modelos foi treinado por 30 épocas, utilizando tamanho de janela variando de 30 a 50 “palavras” para acomodar palavras com grande número de silabas. O tamanho do vetor de representação *vector_size* foi escolhido como 300 conforme exemplos oficiais do algoritmo Word2Vec para garantir que há um número grande o suficiente de características para distinguir as palavras do vocabulário. Todos os modelos Word2Vec apresentados foram treinados utilizando *skip-gram*. Algumas palavras com características específicas relacionadas à representação de etiquetagem foram escolhidas para a comparação dos resultados obtidos através dos diferentes

most_similar(“bebedouro”)					
Modelo Básico - Silabas		**Modelo Básico - Fonética**		**Modelo Básico - Etiquetas**	
Palavra	Similaridade	Palavra	Similaridade	Palavra	Similaridade
bebedouros	0.9704	bebedouros	0.9775	matadouro	0.9768
bebedeira	0.9175	bebedeiras	0.9367	matadeiro	0.9747
cabeça-dura	0.9065	bebedeira	0.9273	motoqueiro	0.9727
bebedeiras	0.9062	belem	0.9016	caldeira	0.9710
cabeçudos	0.9045	zombeteiro	0.8999	barraqueiro	0.9704

most_similar(“bebedor”)					
Modelo Básico - Silabas		**Modelo Básico - Fonética**		**Modelo Básico - Etiquetas**	
Palavra	Similaridade	Palavra	Similaridade	Palavra	Similaridade
bebedores	0.9553	bebedores	0.9535	suportar	0.9521
sabedor	0.9491	liberador	0.9508	definidor	0.9463
liberador	0.9470	recebedor	0.9461	apaziguador	0.9451
barbeador	0.9403	barbeador	0.9428	sinalizador	0.9439
beria	0.9323	barbeadores	0.9343	classificador	0.9438

most_similar(“bebida”)					
Modelo Básico - Silabas		**Modelo Básico - Fonética**		**Modelo Básico - Etiquetas**	
Palavra	Similaridade	Palavra	Similaridade	Palavra	Similaridade
bebido	0.9838	bebido	0.9788	badeco	0.9576
bebia	0.9833	bebidos	0.9776	apagadas	0.9562
bebidas	0.9687	bebidas	0.9759	inadequado	0.9560
bebidos	0.9677	embebida	0.9603	palatabilidade	0.9550
bebi	0.9668	embebidos	0.9588	isocorica	0.9525

most_similar(“nadando”)					
Modelo Básico - Silabas		**Modelo Básico - Fonética**		**Modelo Básico - Etiquetas**	
Palavra	Similaridade	Palavra	Similaridade	Palavra	Similaridade
dando	0.9265	arrecadando	0.9353	tando	0.9854
invalidando	0.9173	nadar	0.9262	pequim	0.9851
lidando	0.9114	andando	0.9141	abanto	0.9833
validando	0.9113	lidando	0.9112	tento	0.9823
apelidando	0.9096	bradando	0.9109	tombo	0.9822

most_similar(“lago”)					
Modelo Básico - Silabas		**Modelo Básico - Fonética**		**Modelo Básico - Etiquetas**	
Palavra	Similaridade	Palavra	Similaridade	Palavra	Similaridade
gola	0.9824	lagos	0.9561	agalega	0.9422
degolado	0.9700	ustilago	0.9363	assinara	0.9363
agolada	0.9687	filago	0.9362	ferreto	0.9362
degolada	0.9565	alaga	0.9291	corados	0.9291
angolanas	0.9462	laga	0.9178	paludo	0.9178

Table 4.1: Comparação entre os resultados da detecção de palavras similares obtidas para os modelos Word2Vec treinados utilizando a estrutura do corpo de texto básico (sem anotações) para cada uma das representações apresentadas.

modelos. Os resultados de detecção de palavras similares para este conjunto de palavras é apresentado a seguir.

Na Tabela 4.1, apresentamos as palavras similares obtidas após os modelos das diferentes representações ortográficas/fonéticas serem treinados utilizando a primeira estratégia de geração de corpo de texto de treinamento apresentada. Na Tabela 4.1a, conseguimos observar esses resultados para a palavra "bebedouro". É possível notar que para a representação de silabas, algumas palavras ortograficamente similares foram encontradas, como "bebedouros", "bebedeira" e "bebedeiras". Apesar disso, o modelo também indicou que certas palavras com pouca relação semântica ou ortográfica fossem detectadas como similares, como no caso de "cabeça-dura" e "cabeçudos". O mesmo comportamento se repete para a representação fonética: temos palavras similares como "bebedouros", "bebedeiras" e "bebedeira", mas em contrapartida foram destacadas palavras com pouca relevância como "belem" e "zombeteiro". O mesmo comportamento se mantém para outras palavras nestes dois modelos (silabas e fonética), como evidenciado pelas Tabelas 4.1b, 4.1c, 4.1d e 4.1e. Uma observação que pode ser feita em relação a este problema é que, como na versão *skip-gram* do algoritmo Word2Vec as palavras ao redor da palavra alvo são consideradas dentro de uma janela de tamanho *n* como palavras de mesmo contexto, se uma palavra possui várias silabas repetidas, a quantidade dessas silabas influenciará na representação da palavra no espaço vetorial, fazendo com que exista um maior nível de semelhança entre esta palavra e outras palavras que possuam tal silaba, como é o caso para a palavra "bebedouro", que só possui a silaba "be" em comum com as palavras "cabeça-dura" e "cabeçudos". Uma outra característica importante a ser mencionada sobre esta modelagem é que, conforme observado na Tabela 4.1e, os modelos são capazes de identificar palavras que possuem as mesmas silabas mas em ordens completamente diferentes. Nesta tabela, para "lago", a palavra "gola" foi considerada uma palavra semelhante para as representações de sílaba e fonética. Considerando o contexto da proposta da representação baseada na etiquetagem, o objetivo é encontrar palavras com semelhança ortográfica ou fonética, o que não necessariamente é o caso para algumas destas palavras.

É importante observar que para o modelo treinado utilizando a representação das etiquetas, o resultado das palavras apresentadas na Tabela 4.1 não parecem similares a primeira vista apesar do alto grau de similaridade. Isso se dá pelo fato de que a representação pela etiquetagem proposta denota características que não são facilmente visíveis, conforme apresentado no Capítulo 2, o que é uma das vantagens dessa representação, proporcionando a capacidade de realizar o agrupamento entre características ortográficas e fonéticas mais genéricas de cada palavra. Desta forma, as palavras que surgem como as mais similares neste modelo são palavras com representações similares aos da palavra alvo. Na tabela 4.1c, temos que "badeco" foi considerada uma palavra similar a "bebida". Neste caso, a representação das duas palavras é exatamente a mesma: [(SA O)(SN V)]1.[(SA O)(SN V)]3.[(SA O)(SN V)]0. Apesar disso, o modelo treinado utilizando essa representação também apresenta os mesmos problemas descritos para as representações de sílabas e fonética, visto que o Word2Vec não considera a ordem das sílabas e a duplicação de "palavras" no contexto.

A Tabela 4.2 apresenta uma comparação entre duas estratégias de modelagem do corpo de texto de treinamento: a estratégia básica de frases simples e a estratégia utilizando um contador relativo a cada uma das silabas. Nas tabelas a seguir, a comparação é realizada apenas para o foco do trabalho, sendo este a representação baseada na etiquetagem. Esta segunda estratégia tem como propósito tentar diminuir a influência de silabas repetidas na representação dos vetores de cada palavra, tentando evitar que palavras com muitas silabas iguais fiquem no topo ao calcular a semelhança com palavras com as mesmas silabas. Desta forma, com o prefixo sendo um contador

most_similar(“bebedouro”)			
Modelo Básico - Etiquetas		**Modelo Contador - Etiquetas**	
Palavra	Similaridade	Palavra	Similaridade
matadouro	0.9768	tabaqueira	0.9588
matadeiro	0.9747	debateu	0.9547
motoqueiro	0.9727	desocupou	0.9532
caldeira	0.9710	bacabeira	0.9508
barraqueiro	0.9704	catapultou	0.9476

most_similar(“bebedor”)			
Modelo Básico - Etiquetas		**Modelo Contador - Etiquetas**	
Palavra	Similaridade	Palavra	Similaridade
suportar	0.9521	batucar	0.9342
definidor	0.9463	demodulador	0.9325
apaziguador	0.9451	coquetel	0.9281
sinalizador	0.9439	decoberta	0.9275
classificador	0.9438	dissipador	0.9273

most_similar(“bebida”)			
Modelo Básico - Etiquetas		**Modelo Contador - Etiquetas**	
Palavra	Similaridade	Palavra	Similaridade
badeco	0.9576	dedica	0.9291
apagadas	0.9562	colodio	0.9214
inadequado	0.9560	parado	0.9194
palatabilidade	0.9550	calita	0.9173
isocorica	0.9525	querubes	0.9156

most_similar(“nadando”)			
Modelo Básico - Etiquetas		**Modelo Contador - Etiquetas**	
Palavra	Similaridade	Palavra	Similaridade
tando	0.9854	pango	0.9761
pequim	0.9851	pudenda	0.9677
abanto	0.9833	equante	0.9676
tento	0.9823	tongo	0.9671
tombo	0.9822	tonga	0.9671

most_similar(“lago”)			
Modelo Básico - Etiquetas		**Modelo Contador - Etiquetas**	
Palavra	Similaridade	Palavra	Similaridade
agalega	0.9422	evaporiticos	0.9268
assinara	0.9282	carriarico	0.9218
ferreto	0.9245	requerida	0.9205
corados	0.9227	punhalada	0.9182
paludo	0.9212	polido	0.9180

Table 4.2: Comparação entre os resultados da detecção de palavras similares obtidas para os modelos Word2Vec treinados utilizando a estrutura do corpo de texto básico (sem anotações) e a estratégia que utiliza os contadores para cada silaba distinta, com os resultados de ambos modelos treinados sendo apresentados apenas para a representação por etiquetagem.

de quantas vezes a silaba aconteceu até aquela posição na divisão de silabas, garantimos que a ordem destas silabas seja parcialmente levada em consideração.

Na Tabela 4.2a, observa-se que as palavras mais similares não se mantiveram as mesmas com a adição do contador. O principal motivo pelo qual isso se dá é que, apesar de silabas repetidas (neste caso, as etiquetas das silabas) aumentarem o peso daquela silaba no contexto da palavra, estas não são consideradas como informações distintas no Modelo Básico. Ao adicionar o contador como prefixo no Modelo Contador, cada silaba repetida torna-se única, visto que o contador não repete, facilitando identificar palavras mais semelhantes quando silabas repetem. Para a palavra "bebedouro" apresentada na Tabela 4.2a, temos nos dois modelos as palavras "matadouro" e "tabaqueira" como as palavras mais semelhantes. A representação de "bebedouro" na etiquetagem é dada por [(SA O)(SN V)]1.[(SA O)(SN V)]1.[(SA O)(CN1 V)(CN2 V)]3.[(SA L)(SN V)]0. Uma observação importante é que as primeiras duas silabas nesta representação (os dois primeiros pares de colchetes) são idênticas, mas independente desta informação, o modelo apresenta "matadouro", representada por [(SA N)(SN V)]1.[(SA O)(SN V)]1.[(SA O)(CN1 V)(CN2 V)]3.[(SA L)(SN V)]0 como a opção mais semelhante, enquanto que, como o Modelo Contador indica, uma opção mais semelhante seria a palavra "tabaqueira", representada por [(SA O)(SN V)]1.[(SA O)(SN V)]1.[(SA O)(CN1 V)(CN2 V)]3.[(SA L)(SN V)]0, a exata mesma representação de "bebedouro". Note que a primeira sílaba (primeiro par de colchetes) de "matadouro" não está presente em "bebedouro", o que deixa mais claro que a representação idêntica a "bebedouro" dada para "tabaqueira" seria a opção correta, mostrando que esta estratégia de modelagem do corpo de texto para treinamento foi mais efetiva levando em consideração a representação baseada na etiquetagem. É possível identificar o mesmo fenômeno nas Tabelas 4.2b, 4.2c, 4.2d e 4.2e. Apesar disso, esta estratégia ainda não utiliza a informação da posição das silabas da palavra no treinamento do modelo, deixando a desejar para identificar palavras que possuem sub-palavras semelhantes.

A Tabela 4.3 apresenta a comparação dos resultados obtidos para o modelo básico e o modelo utilizando a estratégia de modelagem de corpo do texto de treinamento que inclui informações de sub-palavras geradas por cada possível conjunto de sílabas que preserva a ordem das sílabas na palavra. O objetivo de tal método é garantir que a ordem da representação das sílabas influencie de certa forma no treinamento e criação dos vetores de representação das palavras do vocabulário. Como cada conjunto de silabas é gerado como uma nova palavra adicionada à uma frase, as informações de sub-palavras são consideradas como palavras distintas dentro da janela de palavras de contexto da palavra alvo, permitindo também a detecção de palavras semelhantes que contém a palavra alvo ou vice-versa.

È possível notar a influência de tal modelagem do corpo de texto diretamente na Tabela 4.3d, que apresenta os resultados obtidos para a palavra "nadando". No modelo básico, palavras de distintos tamanhos são consideradas similares a "nadando", enquanto diversas palavras com o mesmo número de sílabas e representações idênticas existem no vocabulário. Tais palavras são indicadas como possuindo maior nível de similaridade no modelo que utiliza as informações de sub-palavras no contexto durante o treinamento. Como podemos ver, a representação de "nadando" e "macanga" é a mesma: [(SA N)(SN V)]1.[(SA O)(SN V)]3.[(SA O)(SN V)]0, o que não é verdade para "tando". Resultados similares são apresentados para as Tabelas 4.3a, 4.3b, 4.3c e 4.3e, com o número de silabas das palavras semelhantes sendo mais próximo da palavra alvo.

A Tabela 4.4 apresenta a comparação do modelo básico com um outro modelo utilizando uma combinação das duas modelagens de corpo de texto para as quais os resultados foram apresentados anteriormente. É possível perceber nas Tabelas 4.4a, 4.4b, 4.4c, 4.4d e 4.4e que o conjunto de palavras apresentadas como semelhantes para cada uma das palavras aqui

most_similar(“bebedouro”)			
Modelo Básico - Etiquetas		**Modelo Sub-palavras - Etiquetas**	
Palavra	Similaridade	Palavra	Similaridade
matadouro	0.9768	batedeira	0.9888
matadeiro	0.9747	bacabeira	0.9871
motoqueiro	0.9727	batuqueiro	0.9868
caldeira	0.9710	bebedeira	0.9863
barraqueiro	0.9704	tabaqueira	0.9818

most_similar(“bebedor”)			
Modelo Básico - Etiquetas		**Modelo Sub-palavras - Etiquetas**	
Palavra	Similaridade	Palavra	Similaridade
suportar	0.9521	tucapel	0.9846
definidor	0.9463	batatal	0.9842
apaziguador	0.9451	batucar	0.9829
sinalizador	0.9439	dedicar	0.9825
classificador	0.9438	catecol	0.9823

most_similar(“bebida”)			
Modelo Básico - Etiquetas		**Modelo Sub-palavras - Etiquetas**	
Palavra	Similaridade	Palavra	Similaridade
badeco	0.9576	badeco	0.9854
apagadas	0.9562	bebeto	0.9849
inadequado	0.9560	bacopa	0.9835
palatabilidade	0.9550	batota	0.9833
isocorica	0.9525	bacaba	0.9827

most_similar(“nadando”)			
Modelo Básico - Etiquetas		**Modelo Sub-palavras - Etiquetas**	
Palavra	Similaridade	Palavra	Similaridade
tando	0.9854	macanga	0.9904
pequim	0.9851	mutombo	0.9891
abanto	0.9833	macondo	0.9889
tento	0.9823	mutamba	0.9886
tombo	0.9822	metendo	0.9884

most_similar(“lago”)			
Modelo Básico - Etiquetas		**Modelo Sub-palavras - Etiquetas**	
Palavra	Similaridade	Palavra	Similaridade
agalega	0.9422	lego	0.9688
assinara	0.9282	liga	0.9654
ferreto	0.9245	lapa	0.9632
corados	0.9227	leque	0.9630
paludo	0.9212	lado	0.9629

Table 4.3: Comparação entre os resultados da detecção de palavras similares obtidas para os modelos Word2Vec treinados utilizando a estrutura do corpo de texto básico (sem anotações) e a estratégia que utiliza as informações de sub-palavras para cada grupo de silabas que pode ser formado, com os resultados de ambos modelos treinados sendo apresentados apenas para a representação por etiquetagem.

most_similar(“bebedouro”)			
Modelo Básico: Etiquetas		**Modelo Sp + C: Etiquetas**	
Palavra	Similaridade	Palavra	Similaridade
matadouro	0.9768	bacabeira	0.9903
matadeiro	0.9747	bebedeira	0.9900
motoqueiro	0.9727	batuqueiro	0.9892
caldeira	0.9710	batedeira	0.9877
barraqueiro	0.9704	tabaqueira	0.9856

most_similar(“bebedor”)			
Modelo Básico: Etiquetas		**Modelo Sp + C: Etiquetas**	
Palavra	Similaridade	Palavra	Similaridade
suportar	0.9521	batucar	0.9852
definidor	0.9463	batedor	0.9836
apaziguador	0.9451	batatal	0.9825
sinalizador	0.9439	dedicar	0.9819
classificador	0.9438	tabocal	0.9811

most_similar(“bebida”)			
Modelo Básico: Etiquetas		**Modelo Sp + C: Etiquetas**	
Palavra	Similaridade	Palavra	Similaridade
badeco	0.9576	baguete	0.9826
apagadas	0.9562	babado	0.9824
inadequado	0.9560	batida	0.9819
palatabilidade	0.9550	titica	0.9819
isocorica	0.9525	bebeto	0.9815

most_similar(“nadando”)			
Modelo Básico: Etiquetas		**Modelo Sp + C: Etiquetas**	
Palavra	Similaridade	Palavra	Similaridade
tando	0.9854	macondo	0.9888
pequim	0.9851	micondo	0.9887
abanto	0.9833	mutondo	0.9881
tento	0.9823	medindo	0.9881
tombo	0.9822	mutinga	0.9878

most_similar(“lago”)			
Modelo Básico: Etiquetas		**Modelo Sp + C: Etiquetas**	
Palavra	Similaridade	Palavra	Similaridade
agalega	0.9422	leto	0.9818
assinara	0.9282	lida	0.9801
ferreto	0.9245	laga	0.9793
corados	0.9227	legua	0.9789
paludo	0.9212	loco	0.9779

Table 4.4: Comparação entre os resultados da detecção de palavras similares obtidas para os modelos Word2Vec treinados utilizando a estrutura do corpo de texto básico (sem anotações) e a estratégia que junta as duas estratégias que adicionam maiores informações no contexto: contador relativo a silaba e sub-palavras geradas dos subconjuntos de silabas, com os resultados de ambos modelos treinados sendo apresentados apenas para a representação por etiquetagem.

apresentadas teve pouca variação quando comparado ao modelo anterior. Apesar disso, é possível notar que, quando comparado ao modelo básico, ambos os problemas apresentados anteriormente para este modelo foram solucionados, garantindo que a representação por etiquetagem consiga agrupar palavras que possuem representações similares com maior precisão e levando em consideração as características de cada palavra.

Por fim, a Tabela 4.5 apresenta para cada uma das representações apresentadas anteriormente uma comparação dos resultados obtidos para os modelos treinados utilizando a última estratégia de modelagem de corpo de texto de treinamento, incluindo informação dos contadores de silabas e sub-palavras no contexto de cada palavra do vocabulário. É possível notar que tal estratégia também melhorou a detecção de palavras similares para as representações de silabas e fonética, com palavras mais semelhantes sendo apresentadas para cada um dos modelos.

Apesar disso, é importante notar que o modelo final utilizando a representação de etiquetagem permite um agrupamento de palavras completamente diferente dos apresentados pelas outras duas representações, levando em consideração características ortográficas e fonéticas das palavras que não são facilmente visíveis no alfabeto comum ou no alfabeto fonético internacional.

4.2.1 Discussão sobre resultados

Uma observação importante a ser realizada é que houve grande sucesso em agrupar palavras com estruturas similares na representação das etiquetas utilizadas, sendo possível identificar palavras com estruturas similares, permitindo diversas aplicações de um modelo deste tipo, como a utilização deste para identificar palavras que certos alunos do ambiente escolar possam vir a ter dificuldade com base em palavras de erros já conhecidos.

Uma limitação do Word2Vec é que este algoritmo não gera representações para palavras fora do vocabulário, não permitindo a identificação de palavras semelhantes a palavras não presentes no vocabulário. Apesar disso, dada a nossa modelagem do corpo de texto sendo a representação das palavras em silabas para que o nosso "contexto" fosse as características silábicas da palavra, uma alternativa para a detecção de palavras similares a palavras fora do vocabulário é possível. A biblioteca *gensim* utilizada para gerar estes modelos possui uma função para os modelos Word2Vec que possibilita fornecer um conjunto de palavras de contexto que formam uma frase e obter uma lista de possíveis palavras alvo que podem se encaixar naquele contexto, similar ao processo de treinamento da arquitetura CBOW. Por exemplo, para encontrar uma palavra qualquer não presente no vocabulário, convertemos a palavra para a representação das etiquetas e fornecemos suas silabas (e possivelmente sub-palavras) como o contexto. Como os modelos construíram a representação vetorial com base nessas informações de contexto silábicas, o modelo é capaz de sugerir palavras que se adequam a aquele contexto, efetivamente sugerindo palavras similares a palavras não presentes no vocabulário. Apesar disso, a implementação desse método na biblioteca *gensim* não ofereceu resultados muito satisfatórios, e por restrições de tempo no desenvolvimento deste trabalho, essa via não foi explorada.

Como alternativa para a identificação de palavras fora do vocabulário, optamos por treinar modelos utilizando a implementação do algoritmo FastText da biblioteca, visto que modelos FastText são capazes de gerar representações para palavras fora do vocabulário para buscas de palavras similares. Entretanto, como explicado no Capítulo 3, modelos baseados no FastText utilizam *n-grams* das palavras do vocabulário no processo de treinamento, com os vetores finais sendo influenciados por características das "sub-palavras" de cada palavra. Para encontrar palavras semelhantes a palavras fora do vocabulário, o modelo estima um vetor para a palavra não pertencente ao vocabulário utilizando apenas os *n-grams* gerados a partir desta, visto que não há informação de contexto. Dada nossa modelagem do corpo de texto, elaborada para que as informações silábicas influenciassem diretamente na representação vetorial, as informações de

most_similar(“bebedouro”)					
Modelo Sp + C: Silabas		**Modelo Sp + C: Fonética**		**Modelo Sp + C: Etiquetas**	
Palavra	Similaridade	Palavra	Similaridade	Palavra	Similaridade
bebedouros	0.9780	bebedouros	0.9902	bacabeira	0.9903
bebedeiras	0.8973	bebedeiras	0.9203	bebedeira	0.9900
bebedeira	0.8957	bebedeira	0.9171	batuqueiro	0.9892
beberrao	0.8902	beberrao	0.8928	batedeira	0.9877
duradouro	0.8707	bebedor	0.8902	tabaqueira	0.9856

most_similar(“bebedor”)					
Modelo Sp + C: Silabas		**Modelo Sp + C: Fonética**		**Modelo Sp + C: Etiquetas**	
Palavra	Similaridade	Palavra	Similaridade	Palavra	Similaridade
bebedores	0.9230	bebedores	0.9299	batucar	0.9852
bebedeiras	0.9121	beberrao	0.9216	batedor	0.9836
sabedor	0.9114	sabedor	0.9210	batatal	0.9825
bebedeira	0.9037	bebedeira	0.9186	dedicar	0.9819
beberrao	0.9034	bebedeiras	0.9173	tabocal	0.9811

most_similar(“bebida”)					
Modelo Sp + C: Silabas		**Modelo Sp + C: Fonética**		**Modelo Sp + C: Etiquetas**	
Palavra	Similaridade	Palavra	Similaridade	Palavra	Similaridade
embebida	0.9654	embebida	0.9502	baguete	0.9826
bida	0.9420	bebidos	0.9497	babado	0.9824
bebi	0.9408	bebi	0.9378	batida	0.9819
bebido	0.9393	bebidas	0.9361	titica	0.9819
bebia	0.9336	bebido	0.9358	bebeto	0.9815

most_similar(“nadando”)					
Modelo Sp + C: Silabas		**Modelo Sp + C: Fonética**		**Modelo Sp + C: Etiquetas**	
Palavra	Similaridade	Palavra	Similaridade	Palavra	Similaridade
dando	0.9173	dando	0.8986	macondo	0.9888
findando	0.8947	fundando	0.8936	micondo	0.9887
fundando	0.8935	lidando	0.8932	mutondo	0.9881
sondando	0.8908	sondando	0.8926	medindo	0.9881
andando	0.8894	mandando	0.8910	mutinga	0.9878

most_similar(“lago”)					
Modelo Sp + C: Silabas		**Modelo Sp + C: Fonética**		**Modelo Sp + C: Etiquetas**	
Palavra	Similaridade	Palavra	Similaridade	Palavra	Similaridade
gola	0.9429	filago	0.9387	leto	0.9818
filago	0.9316	lagos	0.9032	lida	0.9801
vedelago	0.9052	medolago	0.8983	laga	0.9793
medolago	0.9025	ustilago	0.8797	legua	0.9789
ustilago	0.8931	vedelago	0.8698	loco	0.9779

Table 4.5: Comparação entre os resultados da detecção de palavras similares obtidas para os modelos Word2Vec treinados utilizando a estratégia final para modelagem do corpo de texto de treinamento, para cada uma das representações apresentadas.

contexto são extremamente importantes para encontrar palavras semelhantes. Como a palavra fora de vocabulário só possui a informação dos *n-grams*, os resultados foram pouco satisfatórios para estas palavras. A utilização dos *n-grams* também influenciou negativamente nos resultados obtidos para palavras dentro do vocabulário, visto que *n-grams* não necessariamente representam sílabas.

5 CONCLUSÃO

Esse trabalho apresentou uma comparação entre diferentes representações ortográficas e fonéticas baseadas em sílabas, com diferentes modelos sendo treinados utilizando os algoritmos Word2Vec e FastText, com o objetivo de validar a etiquetagem proposta para uso no projeto em desenvolvimento Scriba (Adelaide H. P. Silva, 2022) no contexto de processamento de linguagens naturais, mais especificamente sobre agrupamento de palavras com representações similares. Foram apresentadas três representações (silábica, fonética e etiquetagem) a serem utilizadas pelos modelos treinados, com enfoque em descrever o funcionamento da etiquetagem proposta por Adelaide H. P. Silva (2022). Os algoritmos Word2Vec e FastText utilizados para o treinamento dos modelos foram brevemente apresentados, e as duas diferentes arquiteturas (*skip-gram* e *continuous bag of words*) foram brevemente discutidas. Foram treinados modelos Word2Vec utilizando a arquitetura *skip-gram* para as três diferentes representações de palavras apresentadas neste trabalho em combinação com quatro diferentes estratégias para gerar o corpo de texto de treinamento baseado no vocabulário disponibilizado pelo dataset Aeiouadô (Mendonça and Aluísio, 2014). Os resultados da detecção de palavras similares foram apresentados para cada um dos modelos treinados para um conjunto de palavras escolhidas para demonstrar a influência de cada estratégia. De modo geral, o agrupamento pela representação de etiquetagem sendo validado neste trabalho se provou útil e capaz de agrupar palavras similares em sua estrutura ortográfica e fonética, permitindo possíveis aplicações destes modelos em detecção de erros em palavras com representações similares.

Trabalhos futuros incluem explorar outras formas de incluir informações das sílabas nas representações vetoriais das palavras do vocabulário dentro do modelo para melhorar a precisão dos agrupamentos de palavras similares, visto que o algoritmo Word2Vec considera a unidade básica como palavras completas, desconsiderando informações internas da palavra, e o algoritmo FastText considera *n-grams* de caracteres, o que pode não representar corretamente a divisão silábica. Adicionalmente, aplicações de modelos treinados com estas representações podem ser exploradas, especialmente no caso de detecção de erros, como proposto por (Adelaide H. P. Silva, 2022).

REFERENCES

Adelaide H. P. Silva, F. S. (2022). Scriba - uma ia aplicada ao ensino de ortografia.

Association, I. P., Staff, I. P. A., et al. (1999). *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information.

Khan, W., Daud, A., Nasir, J. A., and Amjad, T. (2016). A survey on the state-of-the-art machine learning models in the context of nlp. *kuwait journal of science*, 43.

Ly, A., Uthayasooriyar, B., and Wang, T. (2020). A survey on natural language processing (nlp) and applications in insurance.

Mendonça, G. and Aluísio, S. M. (2014). Using a hybrid approach to build a pronunciation dictionary for brazilian portuguese. In *INTERSPEECH*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space.

Řehřek, R., Sojka, P., et al. (2011). Gensim—statistical semantics in python. *Retrieved from genism. org*.

Zanella, M. S. (2011). Ortografia no ensino fundamental: Um estudo sobre as dificuldades no processo de aprendizagem da escrita. *Poíesis Pedagógica*, 8(2):109–125.