UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE MATEMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

BRUNO FERRAZ DE A. COUTO

AVALIAÇÃO DE ANALOGIAS EM WORD EMBEDDINGS PARA LÍNGUA PORTUGUESA

RIO DE JANEIRO
2020

BRUNO FERRAZ DE A. COUTO

AVALIAÇÃO DE ANALOGIAS EM WORD EMBEDDINGS PARA LÍNGUA PORTUGUESA

Trabalho de conclusão de curso de graduação apresentado ao Departamento de Ciência da Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. João C. P. da Silva

RIO DE JANEIRO

2020

CIP - Catalogação na Publicação

C871a

Couto, Bruno Ferraz de Almeida
Avaliação de Analogias em Word Embeddings para Língua Portuguesa / Bruno Ferraz de Almeida Couto. - Rio de Janeiro, 2020.
47 f.

Orientador: João Carlos Pereira da Silva.
Trabalho de conclusão de curso (graduação) - Universidade Federal do Rio de Janeiro, Instituto de Matemática, Bacharel em Ciência da Computação, 2020.

1. Word Embeddings. 2. Processamento de Linguagem Natural. I. Silva, João Carlos Pereira da, orient. II. Título.

Elaborado pelo Sistema de Geração Automática da UFRJ com os dados fornecidos pelo(a) autor(a), sob a responsabilidade de Miguel Romeu Amorim Neto - CRB-7/6283.

BRUNO FERRAZ DE A. COUTO

AVALIAÇÃO DE ANALOGIAS EM WORD EMBEDDINGS PARA LÍNGUA PORTUGUESA

Trabalho de conclusão de curso de graduação apresentado ao Departamento de Ciência da Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 4 de Agosto de 2020

BANCA EXAMINADORA:

João Carlos Pereira da Silva
D.S.c. (COPPE-UFRJ)

Felipe Fink Grael
M.S.c. (COPPE-UFRJ)

João Antônio R. da Paixão
D.S.c (PUC-RIO)

Dedicatória: À minha família, que me apoiou ao longo da jornada. E à ela, que esteve junto em cada um desses passos.

RESUMO

Estudos na área de Processamento de Linguagem Natural tem indicado o uso de representações vetoriais de palavras e proposto novos modelos de aprendizado de máquina para aprimoramento da geração dessas representações. Essas representações são utilizadas em aplicações de Processamento de Linguagem Natural para substituir a representação textual e utilizar as informações absorvidas pelas representações para melhorar os resultados da aplicação. A avaliação dos modelos treinados é realizada de diferentes formas e são divididas entre formas de avaliação intrínseca e extrínseca. Neste trabalho busca-se explorar as regularidades linguísticas (semântica e sintática) observadas nesses modelos e analisar os resultados do método de avaliação intrínseca em que é aferido a capacidade de resolução de analogias de pares de palavras. Como a maioria dos trabalhos referenciados utilizam da língua inglesa para demonstrar as utilidades de word embeddings, os experimentos foram realizados sobre a língua portuguesa, com intuito de contribuir para os estudos de Processamento de Linguagem Natural e word embeddings no idioma. A acurácia de modelos pré-treinados disponibilizados e dos modelos treinados para este trabalho indicam o potencial de resolução de analogias através dessa técnica. Além disso, a exploração do método de avaliação por analogias expõe particularidades dos resultados obtidos que podem ser enviesadas pela análise da acurácia obtida.

Palavras-chave: Word Embeddings. Avaliação. Processamento de Linguagem Natural.

ABSTRACT

Natural Language Processing studies have indicated the use of word embeddings for text processing tasks and also proposed new machine learning models to improve the embeddings generation. These embeddings are used in Natural Language Processing tasks instead of the token representation to improve the results by using information captured by the vector representation. There are two main categories of methods to evaluate word embeddings: intrinsic and extrinsic. This work explores the linguistic regularities observed in these models and analyses the results of an intrinsic evaluation method using word analogies. Since the majority of works found use word embeddings in english, the objetive of this work is to show the results obtained from portuguese texts and then contribute to Natural Language Processing studies in this language. The accuracy of both pre-trained models and those trained on this work indicates the potential of analogy solving through word embeddings. However, the exploration of this method of evaluation shows some details about how biased the results could be by the obtained accuracy analysis.

Keywords: Word Embeddings. Evaluation. Natural Language Processing

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

PLN	Processamento de Linguagem Natural
NILC	Núcleo Interinstitucional de Linguística Computacional
CI	Case Insensitive
CS	Case Sensitive
Fc	Falha por Cobertura

LISTA DE SÍMBOLOS

$\sim$	Relação

SUMÁRIO

1 INTRODUÇÃO

A área de Processamento de Linguagem Natural (PLN) compõe os estudos computacionais sobre dados na forma textual e desestruturada. O advento de técnicas de aprendizado de máquina contribuíram para que tarefas de PLN pudessem ser realizadas com mais eficiência e com resultados melhores. Como o texto é a forma de dado a ser explorada pela área, é importante que a representação seja a mais otimizada para que grandes conjuntos de dados possam ser processados e gerem informação e resultados melhores. A representação vetorial densa de palavras (*word embeddings*) é uma técnica proposta em trabalhos como (BENGIO et al., 2003), (MIKOLOV et al., 2013) e (PENNINGTON; SOCHER; MANNING, 2014) que permite a representação vetorial das palavras a fim de preservar informações contidas no respectivo contexto através de regularidades estatísticas. Essas representações se destacam por permitirem comparações de similaridade semântica ou sintática entre palavras e demonstraram resultados positivos (MIKOLOV et al., 2013) para a solução de analogias através de operações vetoriais como $\overrightarrow{rei} - \overrightarrow{homem} + \overrightarrow{mulher} = \overrightarrow{rainha}$. A evolução constante dos algoritmos busca aperfeiçoar as informações extraídas com melhores representações e a custos de recurso computacional menores.

A avaliação de modelos de representação vetorial de palavras é realizada de maneiras distintas, como listado em (BAKAROV, 2018). A avaliação por analogias tem certo destaque nesse cenário visto o potencial que soluções semânticas demonstram.

Neste trabalho, buscamos explorar, focados na língua portuguesa, as regularidades linguísticas sugeridas por (MIKOLOV; YIH; ZWEIG, 2013) a partir de modelos pré-treinados disponíveis e observar as diferenças decorrentes dos diferentes processos de treino. A disponibilidade de diferentes modelos em português com um amplo corpus tornou-se possível através do Repositório de Word Embeddings do Núcleo Interinstitucional de Linguística Computacional (NILC). Os algoritmos FastText(BOJANOWSKI et al., 2017) e GloVe(PENNINGTON; SOCHER; MANNING, 2014) possuem diferenças nas suas abordagens/heurísticas, e vamos estudar seus impactos através dos modelos pré-treinados. Além disso, ao comparar a eficácia dos modelos utilizados, temos como objetivo entender como equiparar os resultados de modelos treinados e como a métrica de acurácia poderia mascarar a eficácia de um modelo.

Por fim, realizamos experimentos para treinar a partir de uma base da Wikipedia em português novos modelos e entender como tal processo pode gerar diferentes resultados. Nesse processo, observamos a relevância da fase de pré-processamento, que é uma etapa que prepara a base de dados com limpeza e modificações no corpus para melhor aproveitamento das informações disponíveis. Além disso, tendo visão sobre o corpus utilizado, debatemos como o desempenho é afetado quando palavras não estão presentes no vocabulário do modelo.

O trabalho foi estruturado da seguinte forma: no Capítulo 2 revisamos conceitos e técnicas utilizadas; no Capítulo 3 apresentamos trabalhos relacionados que serviram de referência ou base para os experimentos; no Capítulo 4 realizamos um experimento de exploração de *word embeddings* e benchmark em modelos pré-treinados e outro experimento para avaliação de modelos treinados por nós mesmo; e por fim, no Capítulo 5, detalhamos nossas conclusões a partir dos experimentos e sugestões de trabalhos futuros.

2 REVISÃO DE CONCEITOS

Neste capítulo, definimos os conceitos básicos que são utilizados nos trabalhos referenciados assim como uma base de conhecimento necessária para o entendimento dos experimentos realizados.

2.1 APRENDIZADO DE MÁQUINA

A disciplina de Aprendizado de Máquina estuda os métodos e algoritmos que permitem observar padrões de dados e extrair conhecimento dos mesmos através de uma série de exemplos de forma automatizada. Assim como no funcionamento cognitivo, tais padrões observados fundamentam a ideia de aprendizado da máquina, uma vez que a mesma poderia aplicar o mesmo modelo obtido para identificar novos objetos ou aperfeiçoar o modelo com novas observações e ajustando o padrão. Através desse conhecimento obtido, observamos aplicações em áreas mais abrangentes como Inteligência Artificial, apoiando, por exemplo, processos de tomada de decisão. Como definido em (JAMES et al., 2013), o Aprendizado Estatístico é o conjunto de abordagens para modelar e definir uma função que recrie o mais próximo possível o padrão de dados observado. Em (MITCHIE; SPIEGELHALTER; TAYLOR, 1994), a disciplina de Aprendizado consiste na aplicação de algoritmos para obtenção dos padrões, porém como observado em (JAMES et al., 2013), a disciplina atualmente engloba os estudos e teorias sob Aprendizado Estatístico de forma a não haver mais uma distinção tão grande entre as áreas.

Como definido em (ALPAYDIN, 2016), o objetivo geral da disciplina é obter um modelo que reproduza o comportamento dos dados. E para isso, diferentemente de programas tradicionais, os algoritmos de aprendizado devem ser capazes de ajustar os parâmetros para que o desempenho ótimo seja alcançado. Dessa maneira, uma característica geral dos algoritmos de Aprendizado de Máquina é a iteração sucessiva dos parâmetros, conhecidos por hiper-parâmetros, que ajustam a capacidade do modelo de reproduzir os padrões observados. E entendemos assim, essa iteração como o processo de aprendizado.

Modelos de Aprendizado de Máquina, de forma generalizada, podem ser definidos, conforme os problemas que os descrevem, em processos de aprendizado supervisionado ou não-supervisionado (JAMES et al., 2013). Modelos Supervisionados dispõe de uma base de dados cujos parâmetros observados tem valores esperados de predição. Dessa maneira, o modelo obtido deve ser capaz de, para cada parâmetro X_i prever um resultado Y_i e assim, reproduzir ao máximo possível a esse padrão pré-estabelecido. O Aprendizado Supervisionado costuma ser realizado, a partir da base de dados, através de uma divisão da amostra em um conjunto de treino, que é utilizado para o aperfeiçoamento[1] do modelo,

[1] em inglês, é utilizado o termo *fit* que também pode ser entendido como ajuste

e um conjunto de teste, ao qual é submetido o modelo para avaliação de sua qualidade. A acurácia é uma métrica comum utilizada nesses modelos para avaliar a capacidade do mesmo em prever o padrão em novos conjuntos de dados, isto é, não utilizados no processo de treino. Dispostos os dados de treino, a previsão[2] do modelo deve corresponder ao valor esperado e cada correspondência contribui para a acurácia geral do modelo. Um exemplo de modelo supervisionado é a técnica de classificação, que a partir de um conjunto de dados tem como objetivo classificar cada observação com um identificador esperado.

Modelos não-supervisionados, por sua vez, definem um desafio maior como observado em (JAMES et al., 2013). Para cada parâmetro observado, não há uma associação com um resultado esperado e, dessa maneira, o modelo não tem um correspondente de previsão. Para tal problemática, técnicas estatísticas são utilizadas para definir relações entre os dados observados ou suas variáveis. Uma técnica utilizada, por exemplo, é a de agrupamento cujo objetivo é estabelecer grupos de dados similares e que possam ser interpretados. Diferentemente da técnica de classificação, o agrupamento de dados não dispõe de uma referência dos grupos de observação. Dessa maneira, o número de grupos, por exemplo, tende a variar conforme o problema a ser resolvido e ter uma interpretação mais flexível.

As *word embeddings* que tratamos nesse trabalho são um exemplo de um processo de Aprendizado de Máquina realizado para apoiar outros processos de Aprendizado de Máquina. A geração das representações é realizada através de Redes Neurais Artificiais. Como definido em (ALPAYDIN, 2016), as Redes Neurais são inspiradas pelos estudos do funcionamento cognitivo e são definidas por camadas de unidades de processamento, os neurônios. A organização dos neurônios pode variar conforme a implementação, todavia, de forma simplificada, o objetivo de cada neurônio é validar os dados obtidos dos neurônios conectados. Essa validação consiste em prover para a rede um novo dado a partir da aplicação de pesos, que são definidos para cada neurônio, sobre os dados obtidos. O mecanismo fundamental das Redes Neurais é o aprimoramento dos pesos de cada neurônio de forma a obter, ao fim do processamento, um modelo que reproduza o padrão estudado conforme um parâmetro de entrada.

A partir do processamentos das Redes Neurais, as palavras podem ser representadas de maneira a abstrair informações que a máquina não tem capacidade de absorver como o processamento humano. A representação do dado, como veremos mais nas próximas seções, é uma importante etapa de processos de Aprendizado de Máquina tendo em vista a otimização dos resultados.

[2] em inglês, o termo utilizado é *predict*

2.2 WORD EMBEDDINGS

Um embedding (GOOGLE, 2020) é um termo usado na área de Aprendizado de Máquina para definir um espaço vetorial de baixa dimensão utilizado para representar vetores de alta dimensão. A técnica de geração de embeddings visa representar objetos e extrair atributos importantes que identifiquem os mesmos de forma vetorial . De forma lúdica, podemos entender que cada dimensão do vetor gerado representa um atributo do objeto no espaço, permitindo uma representação distribuída ao longo das dimensões.

Os campos de estudo de Processamento de Imagens e Processamento de Linguagem Natural têm feito uso dessa representação para extrair informações de conjuntos de dados. Um exemplo de aplicação é proposto em (SCHROFF; KALENICHENKO; PHILBIN, 2015), cujo objetivo é o reconhecimento facial de pessoas em um conjunto de imagens. De forma simplificada, o problema é solucionado pela construção de um espaço vetorial através de um conjunto de imagens que contenham faces. Para cada rosto identificado nas imagens, é obtida uma representação vetorial através do processamento das imagens. Uma vez que o espaço seja construído, poderíamos entender de forma abstrata que todas as faces contidas no conjunto de imagens tem uma representação vetorial própria. A cada nova imagem inserida, o processo é refeito de forma a obter representações dos novos rostos no mesmo espaço vetorial construído anteriormente. Dessa maneira, é possível a comparação entre os vetores disponíveis através de uma métrica de semelhança entre ambos. Caso essa semelhança ultrapasse um valor limiar mínimo, pode-se considerar que os vetores representam objetos semelhantes e, portanto, a mesma face.

Os estudos de Processamento de Linguagem Natural (PLN) já encontraram algumas aplicações para as representações de palavras (*word embeddings*), como a tarefa de remoção de ambiguidades (IACOBACCI; PILEHVAR; NAVIGLI, 2016), que detecta se palavras iguais em contextos diferentes podem ter significados distintos. O fundamento, porém, destas representações está na capacidade de extrair informações sintáticas e semânticas das palavras de um texto. Tais propriedades são importantes para entender computacionalmente as relações entre palavras — podendo ser avaliadas de forma análoga aos caso de uso de identificação de rostos, através de operações vetoriais — e obter informações de estruturas maiores, como frases, textos ou documentos.

A representação vetorial de uma palavra está baseada nos conceitos de representações localistas e distributivas. A representação localista consiste em representar cada item através de um descritor único. Por exemplo (EL-AMINE, 2017), considerando que queremos representar as palavras *banana* e *maçã*, poderíamos usar respectivamente os vetores (1,0) e (0,1). Se adicionarmos uma terceira palavra ao conjunto inicial, por exemplo o termo *carro*, teríamos que utilizar um vetor de dimensão 3, onde (1,0,0), (0,1,0) e (0,0,1) representariam, respectivamente, os itens *banana*, *maçã* e *carro*. Essa técnica de representação é conhecida como One-Hot-Encoding. Este exemplo mostra uma das limitações

deste tipo de representação, uma vez que para representar um conjunto de n palavras, precisamos de vetores de dimensão n. Consequentemente, para representar uma sequência de m palavras, precisaríamos de uma matriz com $m.n$ dimensões.

Uma representação distributiva (HINTON; MCCLELLAND; RUMELHART, 1986) busca representar cada item de maneira distribuída através das dimensões consideradas. Neste caso, as palavras *banana*, *maçã* e *carro* poderiam ser representadas pelos vetores (0.2921,0.3327), (0.1132,0.2541) e (0.9921,0.1839). Note que diferentemente do método localista, a dimensão dos vetores não está obrigatoriamente associada a quantidade de itens que temos que representar. Os métodos de representação estudados nesse trabalho fazem uso da representação distributiva porque tal método permite uma noção de similaridade entre as palavras. Retomando o exemplo localista, onde *banana*, *maçã* e *carro* são representadas por (1,0,0), (0,1,0) e (0,0,1), a similaridade de cosseno entre quaisquer dois destes vetores será sempre zero, o que não nos permite estabelecer nenhum tipo de relação entre tais itens. Já no caso da representação distributiva, a similaridade de cosseno entre *banana* e *maçã* (0.954919) é maior que a de *banana* e *carro* (0.78567) e *maçã* e *carro* (0.5666), o que poderia indicar algum tipo de relacionamento entre as duas primeiras palavras. Essa similaridade, de forma subjetiva, pode ser entendida como o fato de ambas serem frutas.

Em (MIKOLOV et al., 2013) é definida uma maneira de transformar representações localistas em distributivas. O método proposto, como veremos adiante em mais detalhe, usa como entrada as representações localistas das palavras e tem como saída da rede neural representações distributivas baseadas nos contextos de cada palavra conforme os algoritmos apresentados no trabalho.

2.3 RESOLUÇÃO DE ANALOGIAS

Considerando o conceito de que cada palavra possui uma representação vetorial, uma abstração que poderíamos considerar é a representação de relações semânticas (referentes aos significados das palavras) ou sintáticas (referentes à função gramatical em uma frase ou sentença) entre palavras através dos vetores associados.

Por exemplo, existe uma relação (gênero) entre as palavras *rei* e *rainha* que gostaríamos de encontrar através da representação vetorial destas palavras. Essa e outras relações poderiam ser expressas através de operações em cima destes vetores. A relação mais usada entre vetores é a similaridade de cosseno. O método obtém a similaridade através do produto interno entre dois vetores $(\overrightarrow{u}, \overrightarrow{v})$ e pode ser avaliado, simplificadamente, como quanto mais próximo de 1, maior a similaridade dos vetores e consequentemente das palavras.

$$\cos(\overrightarrow{u}, \overrightarrow{v}) = \frac{\overrightarrow{u}\,\overrightarrow{v}}{\|\overrightarrow{u}\|\|\overrightarrow{v}\|} = \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\sum_{i=1}^{n} (u_i)^2}\sqrt{\sum_{i=1}^{n} (v_i)^2}} \tag{2.1}$$

Através desta medida, (MIKOLOV et al., 2013) define um método de avaliação de analogias que explora as relações semânticas e sintáticas do modelo distributivo. Como mencionado acima, as palavras *rainha* e *rei* são relacionadas através de uma relação gramatical de gênero. Desta forma, analogamente, *rei* está para *rainha* assim como *homem* está para *mulher*. De maneira abstrata, podemos interpretar que a operação de subtração entre os vetores que representam as palavras *rei* e *rainha* ($\overrightarrow{rei} - \overrightarrow{rainha}$) obteria um vetor de relação entre ambos. De forma análoga, obtemos um vetor de relação entre *homem* e *mulher* ($\overrightarrow{homem} - \overrightarrow{mulher}$).

Conforme (MIKOLOV; YIH; ZWEIG, 2013), espera-se que os vetores de relação obtidos representem o conceito de gênero, e portanto sejam vetores com um alto grau de similaridade. Essa característica, obtida através da operação em exemplo, é observada em (MIKOLOV; YIH; ZWEIG, 2013) como uma regularidade que pode representar uma relação semântica ou sintática entre as palavras. Podemos estruturar esse problema através de uma *questão* (MIKOLOV et al., 2013) que definimos como uma composição de quatro palavras $\{x_1, x_2, y_1, y_2\}$ e dois pares análogos (x_1 está para y_1 ($x_1 \sim y_1$) assim como x_2 está para y_2 ($x_2 \sim x_2$)).

Por exemplo, considerando o conjunto de palavras {*rei*, *homem*,*rainha*, *mulher*}, já observamos que os pares de palavras (*rei*,*rainha*) e (*homem*,*mulher*) guardam entre si uma relação, que neste caso, diz respeito ao gênero.

Podemos representar esta relação entre os respectivos vetores por $\sim$, ou seja, *rei está para rainha* ($\overrightarrow{rei} \sim \overrightarrow{rainha}$) e *homem está para mulher* ($\overrightarrow{homem} \sim \overrightarrow{mulher}$). O que estamos interessados é em usar esta relação de analogias para nos ajudar a determinar outros termos relacionados.

Podemos manipular a relação de forma que, a partir de três palavras, a quarta possa ser obtida.

$$\overrightarrow{x_1} - \overrightarrow{x_2} = \overrightarrow{y_1} - \overrightarrow{y_2} \tag{2.2}$$

$$\overrightarrow{x_1} - \overrightarrow{x_2} + \overrightarrow{y_2} = \overrightarrow{y_1} \tag{2.3}$$

Por similaridade semântica, a operação $\overrightarrow{rei} - \overrightarrow{homem} + \overrightarrow{mulher}$ resultaria em um vetor que teria uma alta similaridade com o vetor $\overrightarrow{rainha}$. Podemos interpretar a operação definida ($\overrightarrow{rei} - \overrightarrow{homem} + \overrightarrow{mulher} = \overrightarrow{rainha}$) de maneira que subtraindo semanticamente *homem* de *rei*, e somado ao resultado *mulher*, obtemos a palavra *rainha*. A analogia {*rei, homem, rainha , mulher*} pode ser extraída independente da composição da questão.

A operação vetorial descrita anteriormente é conhecida como *3CosAdd*, conforme (LEVY; GOLDBERG, 2014), e é a forma de resolução de analogias mais comum. (LEVY; GOLDBERG, 2014) também observa a vulnerabilidade dessa operação a termos dominantes que determinarão a quarta palavra (por exemplo, para a analogia $\overrightarrow{Inglaterra} -$

$\overrightarrow{Londres} + \overrightarrow{Badgá}$, é obtido $\overrightarrow{Mosul}$ em vez $\overrightarrow{Iraque}$ pelo fato de a relação entre $\overrightarrow{Mosul}$ e $\overrightarrow{Badgá}$ predominar visto a alta similaridade entre as palavras). Para contornar esse problema, é detalhado em (LEVY; GOLDBERG, 2014) o método *3CosMul*, que utiliza o fator multiplicativo para evidenciar as relações das palavras.

3 TRABALHOS RELACIONADOS

Neste capítulo, referenciamos trabalhos que contribuíram para a fundamentação de nossos experimentos assim como estudos prévios sobre modelos de *word embeddings*. Detalhamos, de forma simplificada, a evolução dos modelos e enfim observamos dois trabalhos referentes à aplicação para língua portuguesa.

3.1 TREINAMENTO DE WORD EMBEDDINGS

O conjunto de técnicas propostos em (MIKOLOV et al., 2013) para treinamento de modelos de representação vetorial de palavras demonstraram a possibilidade de lidar com elevadas dimensões de dados a um custo e uma complexidade mais tangíveis que trabalhos anteriores. Os modelos propostos seguem uma evolução de uma popular arquitetura proposta em (BENGIO et al., 2003) conhecida por Modelo de Linguagem de Rede Neural, abreviada em inglês para NNLM. A proposta desse modelo, de forma simplificada, era obter a sequência de palavras em uma sentença dada uma palavra como entrada. A saída do modelo consistia em um conjunto de representações das palavras e as probabilidades conjuntas de sequências de palavras que estivessem no conjunto de dados de treino.

As arquiteturas propostas em (MIKOLOV et al., 2013) foram denominadas como Continuous Bag of Words (CBOW) e Continuous Skip-gram. Os algoritmos foram implementados e distribuídos com o nome de *Word2Vec*[1] do Google. Posteriormente, foram propostas evoluções desses métodos baseados em uma janela de contexto, como visto em (BOJANOWSKI et al., 2017). Este trabalho visa enriquecer as representações tendo como alvo não somente as palavras mas também unidades menores, como sílabas, em busca de informação na morfologia das palavras.

Outra abordagem para o treinamento de *word embeddings* é proposta em (PENNINGTON; SOCHER; MANNING, 2014). O objetivo do método é aferir as co-ocorrências globais (*Global Vectors*), diferentemente das janelas de contexto propostas previamente, das palavras e fazer uso das estatísticas provenientes em busca de regularidades linguísticas.

3.1.1 CBOW e Skipgram

Conforme (MIKOLOV et al., 2013), a heurística que fundamenta o CBOW consiste em determinar uma palavra através do contexto no qual ela está inserido. Considerando que o contexto é definido por $2n$ termos ao redor de um alvo, de forma a termos n palavras antes e n depois da palavra desejada, cada item do contexto é utilizada para a predição da palavra do meio. A combinação das representações vetoriais de cada uma dessas $2n$

[1] https://code.google.com/archive/p/word2vec/

palavras é a entrada no algoritmo para a predição da representação vetorial do termo alvo.

O nome do algoritmo faz referência ao método de *bag-of-words* que consiste em uma metodologia de representação de palavras em um conjunto de documentos. O método não considera a ordem dos termos contextuais a fim de obter-se a palavra alvo.

O algoritmo Continuous Skip-Gram (MIKOLOV et al., 2013) inverte a proposta do CBOW ao ter como objetivo obter o contexto mais provável dado uma certa palavra de entrada. O contexto pode ser entendido como um conjunto de palavras limitados a uma distância do termo de entrada. Para uma item de entrada w_i e uma distância N, o algoritmo vai definir um número R de palavras que estarão a uma distancia menor ou igual a N de w_i, seja para frente ou para trás na sentença. E através das camadas da Rede Neural implementada, a representação da palavra alvo é aprimorada a fim de que "acerte"o melhor contexto dado aquela palavra.

Os experimentos de (MIKOLOV et al., 2013) demonstraram que um maior intervalo contextual sugeriria uma melhor qualidade das representações obtidas, uma vez que cada palavra de entrada influenciaria as palavras mais distantes e suas representações seriam enriquecidas. Todavia, quanto mais palavras associadas, maior a complexidade computacional.

Apesar de mais referenciado como apenas Skipgram, o nome provém da técnica homônima que é a base para o algoritmo proposto por (MIKOLOV et al., 2013). Como definido em (GUTHRIE et al., 2006), o Skipgram é uma técnica que adota n-grams, conjuntos de n elementos textuais (fonemas, palavras, silabas, caracteres) adjacentes, e define que a adjacência entre elementos pode ser definida a uma distância n, isto é, elementos podem ser pulados.

Os algoritmos CBOW e Skipgram demonstrados em (MIKOLOV et al., 2013) foram implementados e distribuídos pelo Google como uma ferramenta chamada *Word2Vec*. A ferramenta utiliza como entrada um corpus, que é definido como um conjunto de documentos textuais, e provê como saída as representações vetoriais de todas as palavras encontradas ao longo dos documentos. A implementação dos modelos pelo Word2Vec também permite que modelos pré-treinados sejam enriquecidos posteriormente com outros textos ou dados.

3.1.2 Informações de sub-palavra

A disponibilidade do Word2Vec tornou mais acessível a representação de palavras através de vetores. Conforme a proposta de menor complexidade computacional que observamos anteriormente, projetos da área de Processamento de Linguagem Natural poderiam tirar proveito das representações e estudar os ganhos que tal forma do dado trariam às resoluções. Além disso, evoluções e alternativas aos algoritmos CBOW e Skipgram também

foram ganhando espaço, como por exemplo o FastText[2].

O FastText foi um projeto desenvolvido pela equipe do Facebook e disponibilizado como uma ferramenta que implementa o algoritmo de extração de informação em subpalavras proposto em (BOJANOWSKI et al., 2017). A ferramenta tem como objetivo obter representações de palavras e classificadores de textos através de um corpus de entrada. O algoritmo é tido como uma evolução do Skipgram segundo os autores e consiste na extração de informações contidas na construção das palavras para enriquecer os vetores do modelo. Considerando que a morfologia das palavras pode conter informações semânticas importantes sobre a palavra, (BOJANOWSKI et al., 2017) propõe que o uso de tais informações torna-se valioso para algumas linguagens caracterizadas por extensa morfologia. Idiomas latinos, por exemplo, podem apresentar uma quantidade significante de flexões verbais que seriam representadas por vetores únicos.

O algoritmo Skipgram, como vimos anteriormente, determina que as palavras são os elementos para a composição do contexto de previsão. O modelo, portanto, acaba produzindo uma representação vetorial única para cada palavra. Todavia, informações dentro da própria palavra são perdidas, como apontado em (BOJANOWSKI et al., 2017). O trabalho realizado propõe a adaptação do método do Skipgram para avaliar os n-grams a nível de caracteres. Dessa maneira, a palavra "aventura" avaliada com n=3:

<av, ave, ven, ent, ntu, tur, ura, ra>

é segmentada em tri-grams que por sua vez são representados por vetores. No modelo apresentado, a palavra inteira também é adicionada ao conjunto e replicamos a notação de <,> para espaço vazio no início e fim da palavra a fim de indicar prefixos e sufixos. A partir desse conjunto de tri-grams, uma palavra W é composta pela soma das representações de cada elemento do conjunto.

O trabalho em (BOJANOWSKI et al., 2017) reporta alguns resultados significativos em testes entre alguns idiomas. Alguns experimentos demonstraram que a língua alemã, por exemplo, tem resultados superiores com o método de enriquecimento proposto. O idioma possui uma característica linguística em que palavras são compostas conforme unidades semânticas menores. A palavra em alemão "Tischtennis" significa tênis de mesa. Através do método proposto pelo FastText, a decomposição da palavra realça o valor semântico de unidades menores como "tisch" e "tennis", "mesa" e "tênis" respectivamente. O método, além disso, não tratará as palavras 'Tischtennis' e 'Tennis' inteiramente diferente, conforme observado por (BOJANOWSKI et al., 2017).

A partir dos experimentos em tarefas de similaridade de palavras, (BOJANOWSKI et al., 2017) expõe algumas conclusões importantes: (i) foram observados resultados positivos em idiomas como árabe, alemão e russo, indicando o potencial das linguagens à exploração de informações em subpalavras; (ii) para o inglês, em uma base de palavras

[2] https://fastTextfasttext.cc

menos frequentes, a exploração a nível de n-grams incrementou os resultados; (iii) e por fim, testes realizados com uma base de palavras fora do vocabulário, isto é, para as quais o embedding é gerado apenas pela composição dos vetores dos n-grams de palavras que não estavam no corpus de treinamento, obtiveram vetores pelo menos tão bons quanto se não utilizasse o método.

O FastText demonstra certa instabilidade de resultados em (BOJANOWSKI et al., 2017) entre as diferentes tarefas realizadas, porém figurou entre as ferramentas de melhor desempenho na avaliação por analogias de modelos em português (HARTMANN et al., 2017). O projeto está disponível no *GitHub*[3] e pode ser utilizado diretamente na linguagem Python para manusear modelos pré-treinados ou para o treino de modelos novos. Por essa razão, foi utilizado como ferramenta base para o trabalho.

3.1.3 Global Vectors

O método de *Global Vectors*(PENNINGTON; SOCHER; MANNING, 2014), também conhecido como *GloVe*[4], é a abordagem proposta por pesquisadores da Universidade de Stanford para representação de palavras. O algoritmo busca coletar as estatísticas de co-ocorrência das palavras ao longo do corpus. Para isso, o método propõe a utilização de uma matriz de co-ocorrência M em que uma linha i e coluna j, tal que i,j são menores que o tamanho do vocabulário, obtemos a probabilidade da palavra j ocorrer no contexto de i determinado por uma janela de tamanho n. O processo de treinamento é definido por uma matriz M' iniciada de forma aleatória e, através das iterações de ajuste das representações de cada palavra, tende a corresponder à matriz M.

(PENNINGTON; SOCHER; MANNING, 2014) demonstra resultados de que o GloVe supera a acurácia obtida pelo *Word2Vec* (MIKOLOV et al., 2013) e argumenta que a vantagem do modelo está baseada na visão global das co-ocorrências, tornando o modelo mais acurado para tarefas de PLN.

3.2 AVALIAÇÃO DE MODELOS DE WORD EMBEDDINGS

A representação vetorial de palavras (as word-embeddings) difundiu-se no meio de Processamento de Linguagem Natural como uma ferramenta promissora.Vários novos modelos foram propostos sem que uma forma de avaliação dos modelos tenha sido consolidada, como observado por (BAKAROV, 2018). Cada abordagem acaba por refletir a área de pesquisa e atuação dos profissionais envolvidos assim como o método de experimentação empregado. O levantamento feito por (BAKAROV, 2018) engloba, por exemplo, trabalhos realizados por cientistas cognitivos que se distanciam bastante dos métodos utilizados por engenheiros de aprendizado de máquina e Processamento de Linguagem Natural.

[3] https://github.com/facebookresearch/fastText/
[4] https://nlp.stanford.edu/projects/glove/

Duas categorias de métodos de avaliação são definidos na área de Processamento de Linguagem Natural para modelos de distribuição semântica : as avaliações intrínseca e extrínseca. Ambas englobam um conjunto de métricas e/ou aplicações realizadas sobre os modelos treinados.

A avaliação intrínseca consiste de métodos que buscam as relações internas entre palavras que se identificam com a percepção e classificação humana (a palavra *rei* possui uma relação de domínio semântico com outras palavras tais quais *príncipe*, *rainha* e *reinado*). Métricas como similaridade e vizinhança de palavras são exemplos de características que podem ser extraídas e associadas à avaliação dos modelos.

Métodos intrínsecos expõem um problema qualitativo dos métodos de *word embeddings*. A subjetividade da avaliação é questionada uma vez que as bases de testes precisam ser construídas manualmente e as relações exploradas em um modelo contém um viés atrelado aos criadores dos testes. Uma implicação desse problema é a esparsa difusão de conjuntos de testes, muitas vezes construídos para uma avaliação e, portanto, direcionados para um grupo específico de relações propostas.

Os métodos extrínsecos consistem na avaliação dos modelos através de tarefas de Processamento de Linguagem Natural que utilizem as representações vetoriais das palavras. A avaliação é entendida como o impacto das representações das palavras sobre o processamento de linguagem natural. Como listado por (BAKAROV, 2018), temos exemplos desses ramos que fazem uso das representações vetoriais: Extração de Entidades Nomeadas, Análise de Sentimento e POS-Tagging. A acurácia de cada modelo é baseada em um conjunto de saída esperado para cada aplicação. Conforme (BAKAROV, 2018), é assumido que quando um conjunto de *word embeddings* demonstram bons resultados em uma aplicação, há chances de assim ocorrer em outras aplicações. Esses resultados, diretamente relacionados ao desempenho dessas tarefas são, portanto, utilizados como uma pontuação para a avaliação extrínseca.

A propósito de exemplificação, considere a tarefa de Extração de Entidades Nomeadas. De forma simplificada, o objetivo da tarefa é obter dentro do texto e classificar entidades textuais que representem pessoas, produtos, locais, datas (como Barack Obama, Rio de Janeiro, 2020) entre outros. A definição do problema é um pouco sutil, uma vez que "banco" não é uma entidade nomeada, porém "Banco do Brasil" o é. Parte do trabalho de treinamento de modelos de Extração de Entidades Nomeadas é a avaliação de métricas como a precisão, que demonstram a qualidade do modelo em reconhecer as Entidades Nomeadas presentes. Nesse cenário, as palavras podem ser utilizadas através de seus *embeddings* gerados previamente. E, por fim, o desempenho do modelo de extração de Entidades Nomeadas, refletirá na avaliação dos word embeddings utilizados.

Na proposta de avaliação extrínseca, as relações dentro do vocabulário são avaliadas indiretamente através do impacto nas aplicações das representações das palavras. Dessa maneira, as características extrínsecas ao modelo de representação são avaliadas.

A dificuldade da avaliação extrínseca é a dependência de uma aplicação construída. Para uma grande variedade de tarefas externas, há a disponibilidade de ferramentas que as realizem e reduzam o trabalho de implementação, mesmo que os ajustes e parâmetros associados à execução tendam a variar. Essa dependência acrescenta um processo a mais e, por consequência, o ciclo na geração das representações vetoriais é aumentado.

Em (BAKAROV, 2018) são indicados, para cada uma de suas categorias de avaliação, conjuntos de testes ou benchmarks para os modelos treinados. Para métodos intrínsecos comumente utilizados, observamos que as anotações são geradas artesanalmente. A dependência humana desta tarefa cria uma dificuldade independente da linguagem.

No levantamento de (BAKAROV, 2018), constatamos a quantidade de trabalhos realizados para a língua inglesa. Todavia, quando realizamos o trabalho similar de construção de word-embeddings para a língua portuguesa, há um cenário mais escasso de recursos de avaliação - especificamente de ordem intrínseca. Portanto, se o problema torna-se tão evidente para a língua inglesa com tamanha difusão de conjuntos de teste, a escassez para a língua portuguesa agrava ainda mais o problema de avaliação por métodos de similaridade.

3.3 *WORD EMBEDDINGS* PARA LÍNGUA PORTUGUESA

O trabalho de (MIKOLOV et al., 2013) permitiu um novo panorama para Processamento de Linguagem Natural quando propôs formas mais eficientes no treinamento de modelos distributivos. Todavia, o conteúdo produzido desde então, majoritariamente, é baseado na linguagem inglesa. Mesmo com uma dimensão considerável de material disponível, ainda há poucas referências de word-embeddings em português.

O NILC disponibilizou um repositório[5] de word-embeddings com modelos pré-treinados disponíveis para enriquecimento ou aplicações. O projeto conta com um vasto corpus (1.395.926.282 tokens) que serve de base para o treinamento dos embeddings na língua portuguesa. Os modelos treinados são disponibilizados através dos algoritmos de treino, e recortados conforme número de dimensões. O repositório é base para o trabalho (HARTMANN et al., 2017) que consolida uma avaliação sobre os modelos treinados no processo.

A avaliação é baseada no benchmark produzido por (RODRIGUES et al., 2016) , cujo objetivo é traduzir o modelo de avaliação proposto por (MIKOLOV et al., 2013). O projeto contém um conjunto de questões para teste de analogias no modelo divido em dois segmentos para as variantes europeia e brasileira. Cada conjunto é composto por 14 seções que representam uma relação e são dividas em 5 seções semânticas e 9 sintáticas. Por seção, são listadas as perguntas que são a composição das quatro palavras correspondentes a uma analogia. As relações semânticas testadas são: capitais e países populares, todas as capitais e países, país e moeda, cidade e estados e relações de família de

[5] http://nilc.icmc.usp.br/embeddings

palavras. Para as sintáticas, são usadas: adjetivo para advérbio, antônimos, comparativo, superlativo, infinitivo para presente verbal, adjetivos de nacionalidade, infinitivo para o passado verbal, relação de pluralidade e verbos no plural. Para a variante brasileira, são 17558 perguntas e para a europeia, 17487.

```
: capital-common-countries

Atenas Grécia Bagdá Iraque

Atenas Grécia Bancoque Tailândia
```

Exemplo de seção capital e países populares

As quatro palavras da questão são utilizadas de maneira que a partir de três selecionadas, a quarta pode ser obtida pelo método de similaridade de cossenos.

3.4 CONCLUSÃO

O repositório de *word embeddings* foi o primeiro passo para esse projeto em direção a exploração dos modelos distributivos disponíveis para língua portuguesa. Com um vocabulário largo, algumas questões simples puderam ser testadas (como a analogia {*rei*, *homem*, *rainha*, *mulher*}) e corroboraram com os testes de similaridade entre as respectivas palavras, ainda que em português, diferentemente da versão inglesa em (MIKOLOV; YIH; ZWEIG, 2013).

As performances anotadas e reproduzidas pelos modelos disponíveis em (HARTMANN et al., 2017) também foram utilizadas como referência para os experimentos realizados no próximo capítulo. Segundo a avaliação do trabalho, os modelos treinados com *FastText* e *GloVe* tiveram melhores resultados na resolução de analogias e por esse motivo foram utilizados na etapa de exploração do próximo capítulo. E com a facilidade de utilização da ferramenta do *FastText*, escolhemos como base para os modelos treinados para este trabalho.

4 EXPERIMENTOS

Neste capítulo, detalhamos o processo dos três experimentos realizados para este trabalho. Primeiramente, exploramos alguns modelos pré-treinados com o objetivo de entender as propriedades definidas pelos trabalhos relacionados e observar como elas se comportam para a língua portuguesa. No segundo experimento, comparamos os desempenhos dos modelos pré-treinados reproduzindo a avaliação dos trabalhos originais com foco nos parâmetros de avaliação. Por fim, no terceiro experimento treinamos novos modelos a fim de observar como as etapas do processo de geração de word embeddings influenciam os resultados das avaliações, baseado no benchmark do experimento anterior.

4.1 EXPERIMENTO 1

O primeiro experimento realizado neste trabalho foi aplicar testes de analogia sobre modelos pré-treinados. Considerando que é um método amplamente utilizado para avaliações de modelos, ao longo do experimento reutilizamos conjuntos de analogias de outros trabalhos assim como avaliamos outras elaboradas por nós. Os modelos utilizados foram disponibilizados pelo NILC[1] e foram escolhidos pelo idioma representado ser o português(brasileiro e europeu) e número de palavras agregadas.

A avaliação realizada através da analogia é considerada como uma avaliação intrínseca de teor semântico. A biblioteca utilizada (*gensim*[2]) permite a consulta de uma lista das n palavras (por padrão, n=10) mais similares à pergunta através do método que implementa o cálculo de distância de cossenos proposto previamente:

```
model.most_similar()
```

A pergunta é feita através da decomposição em dois parâmetros: positivo e negativo. Cada parâmetro expressa a influência das palavras na operação das analogias. De forma abstrata, palavras positivas são somadas à relação e as negativas são retiradas. Conforme a operação citada na Seção 2.3:

$$\overrightarrow{x_1} - \overrightarrow{x_2} + \overrightarrow{y_2} = \overrightarrow{y_1} \tag{4.1}$$

podemos observas os termos positivos e negativos da equação que usamos no método. De forma simplificada, o método realiza a operação de similaridade de cossenos entre o vetor obtido no lado esquerdo da equação e as demais palavras do espaço vetorial. As n-palavras mais similares ao vetor obtido são retornadas. E consideramos que nossa palavra alvo na analogia ocupe a posição 1 de maior similaridade.

[1] http://nilc.icmc.usp.br/nilc/index.php/repositorio-de-word-embeddings-do-nilc
[2] documentação disponível em: https://radimrehurek.com/gensim/

Podemos exemplificar esse processo utilizando a analogia *rei, homem, rainha , mulher* , citado na Seção 2.3 e utilizada como exemplo da operação vetorial demonstrada em (MIKOLOV; YIH; ZWEIG, 2013). Considerando como parâmetros do método de similaridade:

```
positivo = [rei, mulher]
negativo = [homem]
```

esperamos um resultado que indicasse *rainha* como melhor resultado, isto é, cuja similaridade fosse a maior dentre as demais palavras. O método representa a operação vetorial :

$$\overrightarrow{rei} - \overrightarrow{homem} + \overrightarrow{mulher} = \overrightarrow{rainha} \tag{4.2}$$

Observamos na Tabela 1 os resultados obtidos através de diferentes modelos de treinamento. Os modelos utilizados foram Fasttext(CBOW e Skipgram) e GloVe, disponibilizados pelo *NILC*[3], e foram escolhidos conforme indicação em (HARTMANN et al., 2017) como os modelos com melhor acurácia. A dimensão 300 de treino também corresponde à mais indicada em média.

Tabela 1 – Comparação entre modelos para $\overrightarrow{rei} - \overrightarrow{homem} + \overrightarrow{mulher}$

Fasttext CBOW		**Fasttext Skipgram**		**GloVe**	
Palavra Alvo	**Similaridade**	**Palavra Alvo**	**Similaridade**	**Palavra Alvo**	**Similaridade**
rainha	0.6863	rainha	0.7449	rainha	0.7193
rei/rainha	0.6846	rainha-regente	0.6796	filha	0.6310
rainha-viúva	0.6582	princesa-regente	0.6528	esposa	0.6273
rainha-mãe	0.6532	esposa	0.6449	princesa	0.6068
rainha-avó	0.6457	princesa	0.6373	isabel	0.5972

Para cada modelo, observamos que a posição máxima é ocupada pela palavra alvo *rainha*, o que configura um acerto no método de resolução de analogias. Como apontado em (HARTMANN et al., 2017), a acurácia geral varia conforme os modelos e a dimensão do espaço vetorial utilizado para treinamentos. É também observada em (HARTMANN et al., 2017) a menor acurácia dos modelos quando utilizam o CBOW como algoritmo de treino, à exceção do Wang2Vec (LING et al., 2015), quando comparados a alternativa do Skipgram com mesma dimensão. Os resultados inferiores indicam como causa a característica do algoritmo CBOW desconsiderar a ordem das palavras do contexto, enquanto que o framework Wang2Vec é o único a considerar esse ordenamento.

A analogia *rei, homem, rainha, mulher* tem uma propriedade específica de conter duas relações semânticas. Assim como previamente analisamos a relação de gênero, observamos que entre as duplas {*rei*, *homem*} e {*rainha*, *mulher*} há uma relação que pode ser entendida como algo real, isto é, ocupar uma posição de realeza. Dada essa observação,

[3] mais informações em: http://nilc.icmc.usp.br/nilc/index.php/repositorio-de-word-embeddings-do-nilc

com a mesma analogia, avaliamos a reorganização da equação da analogia para entender se seria possível extrair tal relação.

$$\overrightarrow{rei} - \overrightarrow{homem} = \overrightarrow{rainha} - \overrightarrow{mulher} \tag{4.3}$$

Para chegarmos a palavra homem ou mulher, modificamos a equação previamente usada invertendo o termo negativo com o alvo.

$$\overrightarrow{rei} + \overrightarrow{mulher} - \overrightarrow{rainha} = \overrightarrow{homem} \tag{4.4}$$

ou

$$\overrightarrow{rainha} + \overrightarrow{homem} - \overrightarrow{rei} = \overrightarrow{mulher} \tag{4.5}$$

Observamos na Tabela 2 os resultados encontrados para essa pergunta:

Tabela 2 – Analogia da relação de realeza - FastText CBOW

$\overrightarrow{\textbf{rei}} + \overrightarrow{\textbf{mulher}} - \overrightarrow{\textbf{rainha}}$		$\overrightarrow{\textbf{rainha}} + \overrightarrow{\textbf{homem}} - \overrightarrow{\textbf{rei}}$	
Palavra Alvo	**Similaridade**	**Palavra Alvo**	**Similaridade**
homem/mulher	0.6432	homem/mulher	0.7166
homem	0.6159	mulher	0.6981
esposo	0.6102	mulher-homem	0.6651
meio-mulher	0.5849	«mulher	0.6586
marido-mulher	0.5848	amulher	0.6527

Tabela 3 – Analogia da relação de realeza - FastText Skipgram

$\overrightarrow{\textbf{rei}} + \overrightarrow{\textbf{mulher}} - \overrightarrow{\textbf{rainha}}$		$\overrightarrow{\textbf{rainha}} + \overrightarrow{\textbf{homem}} - \overrightarrow{\textbf{rei}}$	
Palavra Alvo	**Similaridade**	**Palavra Alvo**	**Similaridade**
homem	0.7047	mulher	0.7244
marido	0.6593	’mulher	0.7244
’esposa	0.6394	amulher	0.6227
pai	0.6084	mulher.	0.6099
filho	0.6038	rapaz	0.6031

Tabela 4 – Analogia da relação de realeza - GloVe 300

$\overrightarrow{\textbf{rei}} + \overrightarrow{\textbf{mulher}} - \overrightarrow{\textbf{rainha}}$		$\overrightarrow{\textbf{rainha}} + \overrightarrow{\textbf{homem}} - \overrightarrow{\textbf{rei}}$	
Palavra Alvo	**Similaridade**	**Palavra Alvo**	**Similaridade**
homem	0.6963	mulher	0.6987
filho	0.6240	rapaz	0.5554
marido	0.5961	ela	0.5488
pai	0.5748	menina	0.5305
jovem	0.5624	mãe	0.5255

Como as Tabelas 3 e 4 demonstram, a relação de realeza é encontrada e extraída nos modelos respectivos. A Tabela 2 não obtém um acerto porém retorna um resultado semanticamente próximo do esperado. E a segunda palavra mais similar em cada pergunta, corresponde à palavra alvo.

Conforme (MIKOLOV; YIH; ZWEIG, 2013), as regularidades linguísticas observadas podem ser representadas pelos deslocamentos entre vetores. A operação $\overrightarrow{homem} - \overrightarrow{mulher}$ utilizada acima é um deslocamento vetorial que representa semanticamente o gênero gramatical. Portanto, se repetirmos a operação para outras analogias, esperamos observar tal regularidade. Utilizamos como referência as relações {*imperador*,*imperatriz*}, {*bombeiro*,*bombeira*}, {*pai*,*mãe*} e {*médica*,*médico*} para avaliar a aplicação do vetor de gênero. Observamos nas Tabelas 5 e 6 que somente a relação {*bombeiro*,*bombeira*} não obtém um resultado correto.

Tabela 5 – Relação Gênero - GloVe 300

$\overrightarrow{\textbf{imperador}} + \overrightarrow{\textbf{mulher}} - \overrightarrow{\textbf{homem}}$		$\overrightarrow{\textbf{bombeiro}} + \overrightarrow{\textbf{mulher}} - \overrightarrow{\textbf{homem}}$	
Palavra Alvo	**Similaridade**	**Palavra Alvo**	**Similaridade**
imperatriz	0.6027	enfermeira	0.4548
esposa	0.5837	funcionária	0.4235
constantino	0.5501	bailarina	0.4101
rainha	0.5360	aposentada	0.3969
filha	0.5177	cozinheira	0.3916

Tabela 6 – Relação Gênero - GloVe 300

$\overrightarrow{\textbf{pai}} + \overrightarrow{\textbf{mulher}} - \overrightarrow{\textbf{homem}}$		$\overrightarrow{\textbf{médica}} + \overrightarrow{\textbf{homem}} - \overrightarrow{\textbf{mulher}}$	
Palavra Alvo	**Similaridade**	**Palavra Alvo**	**Similaridade**
mãe	0.8039	médico	0.5929
filha	0.7902	clínica	0.5417
esposa	0.7575	medicina	0.5322
irmã	0.7374	paciente	0.4908
marido	0.7212	saúde	0.4885

Para ambas as consultas, foram utilizados o modelo GloVe 300.

4.1.1 Exploração de novas relações

O segundo passo do experimento de analogias consiste em explorar outros conceitos de relações semânticas, assim como observamos nas Tabelas 2, 3 e 4 . Dessa forma, os exemplos seriam instâncias desses conceitos e poderíamos observar o alcance do modelo para diversas formas de uma mesma analogia. Em (MIKOLOV et al., 2013), um conjunto de conceitos de ordem sintática e semântica foi criado para avaliar o modelo treinado. Todavia, conforme a produção manual desses conjuntos tende a ser custosa, há um certo limite nos conceitos explorados definidos pelos respectivos criadores. Por esse motivo,

buscamos explorar relações não abrangidas e visualizar o comportamento nos modelos pré-treinados em português.

A relação *país-presidente* foi um caso de exploração em que usamos a operação de processamento de analogias para descobrir elementos na base.

$$pais_1 - presidente_1 = pais_2 - presidente_2 \tag{4.6}$$

Assumindo que sabemos um país e um presidente, exploramos a descoberta de um presidente a partir de seu país. Dessa forma, utilizamos uma equação da seguinte forma:

$$pais_{alvo} + presidente_{conhecido} - pais_{conhecido} = presidente_{alvo} \tag{4.7}$$

A relação país-presidente permite uma multiplicidade dos resultados uma vez que há mais de uma opção possível de presidente associado a um país ou até mesmo mais de um país associado a um nome de presidente, mesmo que não referencie a mesma pessoa. A Tabela 7 exemplifica um resultado tido como correto. A palavra "rajoy"faz referência ao ex-primeiro ministro da Espanha, Mariano Rajoy.

Tabela 7 – $\overrightarrow{merkel} - \overrightarrow{alemanha} + \overrightarrow{espanha}$

Fastext Skipgram		Fastext CBOW		GloVe	
Palavra Alvo	**Similaridade**	**Palavra Alvo**	**Similaridade**	**Palavra Alvo**	**Similaridade**
rajoy	0.7286	merkez	0.7139	aznar	0.5727
aznar	0.6592	merkel.o	0.6644	hollande	0.5118
psoe	0.6200	merkell	0.6039	angela	0.4840
zapatero	0.6065	merkels	0.5969	rajoy	0.4739
aznarez	0.5837	aznarez	0.5965	tsipras	0.4697

Tabela 8 – $\overrightarrow{rajoy} - \overrightarrow{espanha} + \overrightarrow{brasil}$

Fastext Skipgram		Fastext CBOW		GloVe	
Palavra Alvo	**Similaridade**	**Palavra Alvo**	**Similaridade**	**Palavra Alvo**	**Similaridade**
governo.dilma	0.5742	tucano.o	0.5261	temer	0.4079
brasileira.brasília	0.5486	brasil),	0.4973	dilma	0.3592
temer	0.5484	fhc	0.4791	interino	0.3561
brasil	0.5476	tucanês	0.4782	rousseff	0.3505
presidente.dilma	0.5410	tucan	0.4755	fhc	0.3432

Observamos na Tabela 8, entretanto, que o resultado não retorna uma palavra aceitável, isto é, não acerta a analogia. Por mais que a palavra *governo.dilma* apareça na primeira posição e tenha uma relação com a presidente Dilma, a palavra configura o que consideramos uma sujeira na base, visto a composição de duas palavras separadas por um ponto. Outro detalhe que podemos observar no resultado da Tabela 7 que fazemos uma pergunta com a palavra *merkel*, referência a Primeira-Ministra da Alemanha. Todavia, mesmo sendo um cargo diferente de presidente, o resultado parece representar a posição política frente ao país e não necessariamente o cargo. Assim também observamos as

primeiras palavras retornadas pelos modelos FastText Skipgram e GloVe, que retornam *rajoy* e *aznar*, respectivos ex-primeiro-ministros da Espanha. Nos demais resultados, a referência a *hollande* é a única associada a um presidente.

A multiplicidade do conceito, todavia, gera uma problemática para a resolução das analogias uma vez que a base de dados de treino pode ditar qual entidade de presidente será retornada. Para tal pergunta, uma solução para composição de um conjunto de teste é que exista alternativas de pergunta contemplando resultados esperados, como por exemplo, múltiplos pares contendo $(brasil, presidente_x)$

A partir dos insumos de perguntas às quais submetíamos o modelo, observamos outros detalhes do modelo.

Tabela 9 – Comparação entre analogias nos datasets em português e inglês

$\overrightarrow{merkel} - \overrightarrow{alemanha} + \overrightarrow{brasil}$		$\overrightarrow{merkel} - \overrightarrow{germany} + \overrightarrow{brazil}$	
Palavra Alvo	**Similaridade**	**Palavra Alvo**	**Similaridade**
brasil)em	0.5761	Rouseff	0.5680
\x93dilma	0.5755	Roussef	0.5585
brasil)\x94,	0.5744	juncker	0.5372
brasilês	0.5669	Dilma	0.5332
presidenta	0.5624	oboma	0.5172

Os modelos utilizados na Tabela 9 foram o FastText Skipgram 300(NILC) e *wiki-news-300d-1MA* disponibilizado[4] por (MIKOLOV et al., 2018). A comparação demonstra a diferença dos resultados para a mesma pergunta nos idiomas português e inglês. Podemos observar a diferença entre os resultados primeiramente quanto a sujeira encontrada. As três primeiras palavras retornadas pela pergunta em português contém algum tipo de má formação dos tokens. Por outro lado, na pergunta em inglês, notamos que um possível resultado é encontrado nas duas primeiras posições. Esse tipo de sujeira encontrada demonstra uma dispersão que as palavras podem ter uma vez que existam mais de uma forma dela no modelo. A múltipla representação da palavra acarreta que no momento de treinamento, pelo menos dois contextos não foram associados já que a palavra alvo era diferente devido a sujeira. A partir de uma investigação da palavra *\x93dilma*, observamos que foram processadas 601835 ocorrências no corpus, comparado a 928201 ocorrências de *dilma*. O prefixo *\x93* é uma referência ao simbolo "e demonstra o impacto do pré-processamento, uma vez que *\x93dilma* torna-se um vocábulo diferente de *dilma* no treinamento.

Outra observação que pudemos realizar sobre os modelos utilizados é o impacto do pré-processamento. Tomando como exemplo o modelo disponibilizado pelo FastText em inglês treinado a partir da base da Wikipedia, observamos a diferença entre a pergunta realizada em caixa alta e baixa. Além da diferença entre o intervalo de similaridade, [0.67,

[4] disponível em https://fasttext.cc/docs/en/english-vectors.html

Tabela 10 – Comparação entre analogias nos datasets em inglês com diferentes escritas

$\overrightarrow{Merkel} - \overrightarrow{Germany} + \overrightarrow{Brazil}$		$\overrightarrow{merkel} - \overrightarrow{germany} + \overrightarrow{brazil}$	
Palavra Alvo	**Similaridade**	**Palavra Alvo**	**Similaridade**
Rousseff)em	0.7313	Rouseff	0.5680
Roussef	0.6911	Roussef	0.5585
Rouseff	0.6840	juncker	0.5372
Dilma	0.6766	Dilma	0.5332
Lula	0.6736	oboma	0.5172

0.73] e [0.51, 0.56], observamos que quatro resultados da pergunta em caixa alta retornam valores aceitáveis e o primeiro resultado contém sujeira. Considerando que são utilizadas Entidades Nomeadas na pergunta e a base de dados foi obtida através da Wikipedia, entendemos que haja uma substancial diferença entre modelos conforme as etapas de pre-processamento utilizadas.

Como última etapa de exploração, levantamos exemplos de perguntas baseados em conceitos específicos que demonstrem outros tipos de relações encontradas nos modelos. Assim como evidenciado na Tabela 11, esse exemplo demonstra alguns resultados também obtidos conforme a diferença dos resultados entre o modelo disponibilizado em inglês e os modelos em português. Novamente, além da acurácia da questão (neste caso, a resposta esperada era *felino*), a diferença de similaridade também é considerável na resposta.

Tabela 11 – Comparação entre pergunta através de modelo em português e inglês

$\overrightarrow{\textbf{gato}} + \overrightarrow{\textbf{canino}} - \overrightarrow{\textbf{cachorro}}$		$\overrightarrow{\textbf{cat}} + \overrightarrow{\textbf{canine}} - \overrightarrow{\textbf{dog}}$	
Palavra Alvo	**Similaridade**	**Palavra Alvo**	**Similaridade**
caninos	0.6211	feline	0.8340
felino	0.6080	cats	0.7032
gatopardo	0.5799	felines	0.6862
canini	0.5735	Feline	0.6237
canin	0.5718	Cat	0.5961

O último conceito que exploramos foi a relação *país-comida*, considerando os relevantes resultados obtidos para a relação *país-presidente* e o potencial de descoberta entre a entidade país e respectivos aspectos culturais.

Tabela 12 – $\overrightarrow{Japan} - \overrightarrow{USA} + \overrightarrow{burger}$

Palavra Alvo	**Similaridade**
hamburger	0.6304
sushi	0.6181
burgers	0.6127
ramen	0.6113
yakitori	0.6036

Tabela 13 – $\overrightarrow{Japan} - \overrightarrow{sushi} + \overrightarrow{burger}$

Palavra Alvo	**Similaridade**
Germany	0.5900
Canada	0.5562
U.S.A	0.5414
Australia	0.5397
Britain	0.5318

Tabela 14 – $\overrightarrow{Japan} - \overrightarrow{sushi} + \overrightarrow{pizza}$

Palavra Alvo	**Similaridade**
Italy	0.6228
Germany	0.6013
Pizza	0.5716
Canada	0.5570
Australia	0.5476

Os pares da analogia foram construídos manualmente baseado em indicações de comidas típicas. Nenhum dos exemplos mostrados nas Tabelas 12, 13 ou 14 retornou resultados esperados em um modelo em português e por isso mantivemos a exploração no modelo em inglês. Como podemos observar na Tabela 12, o modelo não acerta a palavra *sushi* esperada. Todavia, quando modificamos a composição da pergunta (Tabela 13), centralizamos a relação Japão-sushi e buscamos qual país teria relação análoga a *burger* e descobrimos o país *Alemanha*. diferentemente de *USA* como esperado na Tabela 12. Esse exemplo expõe uma problemática sobre o corretismo de uma analogia, que pode ser direcionada pela perspectiva do criador, enquanto que a base de dados de treinamento reflete outra resposta. Eles compõem uma listagem de pares que serviram de base para um conjunto de testes com analogias montadas para o projeto.

4.1.2 Conclusão do Experimento

O experimento 1 expõem alguns detalhes do pré-processamento do texto que afetam a avaliação dos modelos treinados. A normalização é uma etapa comum em processamento de texto, porém nesse cenário de avaliação ela pode gerar dificuldades. Por outro lado, a normalização do texto pode aumentar os contextos atribuídos a uma representação de palavras visto que diversas formas de escrita de uma mesma palavra serão consolidadas em um token. Além disso, podemos observar nos resultados sujeiras textuais que acabam inflando o vocabulário. A Tabela 9 mostra palavras como *\x93dilma* e *brasil)\x94*, que acabam poluindo o espaço e não contribuindo para a representação das palavras ideais (*brasil* e *dilma*, nesse caso).

4.2 EXPERIMENTO 2

A avaliação com analogias de modelos treinados é representada pela acurácia que o respectivo modelo atinge perante um conjunto de testes. Os trabalhos de (MIKOLOV et al., 2013) e (HARTMANN et al., 2017) são avaliados de forma análoga, uma vez que a avaliação realizada em português é uma tradução do conjunto de testes disponibilizado pelo trabalho em inglês.

O benchmark dos modelos, isto é, uma comparação do desempenho dos modelos na resolução das analogias, permite a visualização de características dos modelos testados. Neste experimento, realizamos um benchmark entre os três modelos utilizados anteriormente (GloVe, Fastext utilizando Skipgram e Fastext utilizando CBOW) e tidos como os mais acurados em (HARTMANN et al., 2017) para a avaliação de analogias. Nosso objetivo é explorar os parâmetros de avaliação e os reflexos nos resultados. Todos os modelos foram treinados com dimensão 300.

Para essa comparação, definimos uma permutação entre os parâmetros de avaliação disponíveis no método utilizado da biblioteca *gensim*. Redefinimos os parâmetros como *falha por cobertura*[5](Fc) e *sensibilidade*(CS-*Case Sensitive*) e *insensibilidade*(CI-*Case Insensitive*). O parâmetro Fc define que o algoritmo deve considerar perguntas cujas palavras não estão presentes no modelo como falhas, produzindo um erro para tal pergunta. Caso contrário, o algoritmo pula a referida pergunta e não afeta a acurácia total. *Sensibilidade* define que o algoritmo deve avaliar se a palavra alvo corresponde exatamente à palavra obtida no processamento da analogia. Caso contrário (SI), o algoritmo normaliza tanto a palavra alvo como a obtida no processamento em caixa alta.

O último parâmetro do método de avaliação é o *vocabulário restrito*. Esse parâmetro determina a quantidade de palavras que compõe o vocabulário utilizado para responder às perguntas. Caso alguma palavra de uma pergunta não esteja no vocabulário restrito, a pergunta é considerada como não encontrada. É possível utilizar o tamanho integral do modelo treinado, porém por questões de recursos e otimização, utiliza-se uma amostra do modelo, com valor padrão 300000. Esse vocabulário, por convenção, reflete as palavras mais frequentes do modelo já que o índice das palavras tende a ser ordenado por frequência pelos modelos atuais. O ajuste do *vocabulário restrito* não foi coberto pelo experimento por causa de escopo.

A configuração padrão do método, utilizado em (HARTMANN et al., 2017), é sem *Falha por Cobertura* e insensitiva (CI).

Os modelos utilizados têm a característica de serem treinados com normalização do texto, fazendo seu vocabulário ser preenchido por palavras em caixa baixa. Todavia, como os benchmarks são criados para avaliar diversos modelos, observamos que na relação *país-capital*, por exemplo, por tratar-se de entidades, cada palavra é representada pela

[5] No método, o parâmetro original é *Dummy4Unknown*, mas foi renomeado por melhor legibilidade

Tabela 15 – Glove 300

Testset	FcCI	CI	FcCS	CS
LX-4WAnalogiesBr	0.43	0.47	0.17	0.42
LX-4WAnalogies	0.42	0.46	0.17	0.43

Legenda: Fc - Falha por cobertura, CI - Insensibilidade, CS - Sensibilidade

Tabela 16 – FastText Skipgram 300

Testset	FcCI	CI	FcCS	CS
LX-4WAnalogiesBr	0.49	0.53	0.25	0.62
LX-4WAnalogies	0.49	0.53	0.25	0.63

Legenda: Fc - Falha por coberturas, CI - Insensibilidade, CS - Sensibilidade

Tabela 17 – FastText CBOW 300

Testset	FcCI	CI	FcCS	CS
LX-4WAnalogiesBr	0.36	0.39	0.24	0.59
LX-4WAnalogies	0.38	0.41	0.25	0.62

Legenda: Fc - Falha por cobertura, CI - Insensibilidade, CS - Sensibilidade

norma própria com a primeira letra em caixa alta somente. Essa particularidade impacta a avaliação dos modelos quando o parâmetro *sensibilidade* é utilizado, uma vez que, por exemplo, a palavra *Atenas* não será encontrada no vocabulário dos modelos. Dessa maneira, a seção inteira acabará tendo um total de 0 de acurácia. O parâmetro de *falha por cobertura* é importante para a avaliação por representar a abrangência do modelo. Quanto menos palavras do conjunto de testes estiverem representadas, menor será a acurácia do modelo. E combinado à particularidade de modelos normalizados, o impacto é ainda maior visto que palavras que compõem a pergunta e que não sejam normalizadas, não estarão representadas no modelo pré-treinado, por fim reduzindo a acurácia geral.

Em contrapartida, quando utilizamos o parâmetro de *sensibilidade*, o impacto é inverso se não utilizado o parâmetro de *falha por cobertura*. Considerando que as palavras próprias não estarão representadas no modelo, a avaliação da acurácia desconsiderará todas as perguntas de capital-cidade. O impacto não é necessariamente positivo porque não garante que as demais perguntas serão acertadas, porém reduz a quantidade de perguntas efetuadas.

A Tabela 18, por fim, expõe um problema da avaliação por analogias não observadas nos benchmarks. Notamos que a cobertura do modelo cai para menos da metade quando aplicado a sensibilidade das palavras em razão das palavras listadas no conjunto de testes com caixa alta. Além disso, mesmo no caso insensitivo, 8,35% das perguntas do benchmark não são efetuadas por não haver representação de alguma palavra na composição.

Os modelos abordados na Tabela 18 foram primeiramente submetidos a um benchmark

Tabela 18 – Cobertura(%) dos modelos pré-treinados (*NILC*)

Teste	CI	CS
LX-4WAnalogiesBr	91.65	40.52
LX-4WAnalogies	92.31	40.74

em (HARTMANN et al., 2017). O benchmark é realizado com as configurações padrão de sensibilidade e falha por cobertura, equivalente à configuração CI das Tabelas 15, 16 e 17. Todavia, o argumento de vocabulário restrito é irrestrito por padrão no script de avaliação[6] utilizado em (HARTMANN et al., 2017). Dessa forma, o modelo inteiro é utilizado para perguntas, o que tende a aumentar a cobertura do modelo (Tabela 19), visto que todas as palavras do corpus de treino são utilizadas e assim diminuindo a chance de alguma palavra do conjunto de teste não ser encontrada. Por outro lado, como a indexação do vocabulário costuma ordenar as palavras por ordem decrescente de frequência, as palavras adicionais pela não restrição incluem palavras de baixa frequência que podem ter representações muito específicas para os poucos contextos em que são encontradas. Considerando que o argumento de falha por cobertura não é utilizado e caso essas palavras adicionais componham uma pergunta que gera uma resposta incorreta, elas afetam negativamente a acurácia geral. No caso restritivo, essas palavras não encontradas seriam neutras na acurácia.

Tabela 19 – Cobertura(%) dos modelos pré-treinados (*NILC*) com vocabulário irrestrito

Teste	CI	CS
LX-4WAnalogiesBr	96.99	41.91
LX-4WAnalogies	97.19	42.13

Tabela 20 – Acurácia de modelos pré-treinados (*NILC*)

Teste	FastText CBOW	FastText Skipgram	GloVe
LX-4WAnalogiesBr	0.30	0.45	0.47
LX-4WAnalogies	0.31	0.45	0.46

Fonte:(HARTMANN et al., 2017)

Como observamos na Tabela 20, os modelos FastText CBOW e Skipgram obtém acurácias menores quando comparados ao benchmark utilizando os parâmetros padrões nas Tabelas 16 e 17. Atribuímos esses resultados ao parâmetro de vocabulário irrestrito utilizado em (HARTMANN et al., 2017). Entendemos que as palavras adicionais incluem palavras componentes de perguntas porém não geram respostas corretas, reduzindo a acurácia dos respectivos modelos.

[6] https://github.com/nathanshartmann/portuguese_word_embeddings

4.2.1 Conclusão do Experimento

Finalmente, os benchmarks demonstram uma visão geral da acurácia dos modelos, porém a manipulação dos parâmetros de avaliação evidenciam características importantes dos modelos e o seus respectivos processos de treino. Além disso, como cada acerto é considerado a partir da resolução correta da quarta palavra da pergunta, notamos que a reordenação da pergunta não é assegurada como um acerto. Dessa forma, podemos também concluir que o método de avaliação utilizado corresponde somente a uma forma de ordenação da pergunta. Para uma efetiva acurácia do modelo em uma analogia, seria necessário replicar no conjunto de teste as demais reordenações da respectiva pergunta.

4.3 EXPERIMENTO 3

A ultima fase de experimentos consiste em treinar novos modelos e explorar as alternativas existentes nesse processo. No fim, realizamos uma avaliação análoga ao experimento anterior sobre os novos modelos treinados.

O primeiro passo consiste na escolha do corpus que define a base de dados para treinamento. O conjunto de treinamento é fundamental pois as representações das palavras dependem diretamente da aparição das mesmas no corpus. A frequência das palavras impactam as respectivas representações uma vez que mais contextos serão associados à representação.

Em seguida temos a fase de pré-processamento do texto selecionado que é uma etapa comum à tarefas de Processamento de Linguagem Natural. Decidimos, portanto, avaliar o impacto do pré-processamento e da dimensão de treino do FastText.

Outros parâmetros[7] estão envolvidos no processo de treinamento com os modelos, porém não foram experimentados por razões de escopo e disponibilidade de recurso. Esses parâmetros estão associados com particularidades do framework FastText, como a janela de n-grams ou a janela de subword information, assim como outros parâmetros, conhecidos como hiper-parâmetros, que otimizam os resultados da Rede Neural.

Para este experimento, escolhemos o framework FastText que é um dos frameworks no Experimento 1. A base de dados utilizada para os modelos treinados é o *dump*[8] de setembro de 2019 da Wikipedia para o idioma em português. Conforme a documentação do FastText, utilizamos um script [9] para reduzir o *dump* em formato .xml removendo as marcações do arquivo e termos como base um arquivo texto somente com os textos. Essa limpeza do arquivo, além de facilitar o processamento do texto, reduz o tamanho total do arquivo de aproximadamente 8GB para 2,7GB.

[7] https://fasttext.cc/docs/en/python-module.html#train_unsupervised-parameters
[8] Disponivel em: https://dumps.wikimedia.org/
[9] Indicado em https://fasttext.cc/docs/en/unsupervised-tutorial.html

Os modelos foram treinados conforme a base de dados sem pre-processamento(noPP) e com pre-processamento composto de: remoção de stopwords e normalização de texto (norm-stop). O treinamento pelo framework do FastText consistiu na variação, para cada uma das bases acima, da variação entre dimensão (100 ou 300) e do algoritmo de treinamento (Skipgram ou CBOW). Os nomes de cada modelo são a composição de cada parâmetro utilizado. Os modelos treinados foram submetidos ao mesmo benchmark utilizado no Experimento 1.

Tabela 21 – Benchmark de modelos CBOW Treinados

	Acurácia	
Modelo	**LX-4WAnalogiesBr**	**LX-4WAnalogies**
noPP-cbow-100	0.4658	0.4963
norm-stop-cbow-100	0.5061	0.5347
noPP-cbow-300	0.5202	0.5549
norm-stop-cbow-300	**0.5405**	**0.5749**

Tabela 22 – Benchmark de modelos Skipgram Treinados

	Acurácia	
Modelo	**LX-4WAnalogiesBr**	**LX-4WAnalogies**
noPP-skip-100	0.4583	0.4696
norm-stop-skip-100	0.4525	0.4560
noPP-skip-300	0.5446	0.5505
norm-stop-skip-300	**0.5487**	**0.5522**

Legenda:

Os dois algoritmos de treinamento obtém seus melhores resultados junto à opção de pré-processamento definida. A remoção de stopwords é uma etapa importante visto que são elementos textuais que aparecem em grande quantidade no corpus. A alta frequência desses tokens tende a impactar a representação das demais palavras da sentença visto que compõem o contexto das palavras que virão a ser treinadas. Observamos que no modelo *noPP-skip-100*, das 100 palavras mais frequentes do modelo 39 estão presentes no conjunto de stopwords a serem extraídas. Acreditamos, portanto, que a remoção das stopwords tem uma influência sobre a representação das demais palavras. A normalização realizada, entretanto, teve um efeito redundante uma vez que não há evidências de palavras em caixa alta no corpus devido ao processamento do script de decupagem de xml utilizado.

O conjunto de testes utilizado define 14 seções de conceitos a serem testados, onde os 5 primeiros são de ordem semântica e os demais, sintáticos. Decidimos observar o impacto em cada segmento para cada modelo utilizando o script de avaliação do FastText[10]. As avaliações em (HARTMANN et al., 2017) indicam o FastText como o modelo que tem

[10] Disponível na seção de exemplos em https://github.com/facebookresearch/fastText/

melhor desempenho nas analogias sintáticas e o GloVe, não utilizado para treinamentos neste experimento, com os melhores resultados para analogias semânticas.

Tabela 23 – Acurácia Semântica

	Acurácia	
Modelo	**LX-4WAnalogiesBr**	**LX-4WAnalogies**
noPP-cbow-100	0.1906	0.1943
norm-stop-cbow-100	0.2049	0.1993
noPP-cbow-300	0.1653	0.1735
norm-stop-cbow-300	0.1637	0.1701
noPP-skip-100	0.3492	0.3383
norm-stop-skip-100	0.3532	0.3324
noPP-skip-300	**0.4143**	**0.3946**
norm-stop-skip-300	0.4029	0.3880

No. Perguntas - BR:8825 - PT:8825

Tabela 24 – Acurácia Sintatica

	Acurácia	
Modelo	**LX-4WAnalogiesBr**	**LX-4WAnalogies**
noPP-cbow-100	0.5901	0.6247
norm-stop-cbow-100	0.6414	0.6761
noPP-cbow-300	0.6803	0.7171
norm-stop-cbow-300	**0.7096**	**0.7455**
noPP-skip-100	0.5076	0.5254
norm-stop-skip-100	0.4970	0.5080
noPP-skip-300	0.6034	0.6168
norm-stop-skip-300	0.6084	0.6186

No. Perguntas - BR:8733 - PT:8662

Primeiramente, podemos constatar na Tabela 23 a baixa acurácia semântica dos modelos CBOW observada em (HARTMANN et al., 2017). Os resultados reforçam a hipótese que o desordenamento do contexto de uma palavra tende a não capturar as regularidades semânticas. Por outro lado, assim como apontado em (HARTMANN et al., 2017) devido à característica morfológica do algoritmo de *subword information*, os modelos obtém acurácias melhores em ordem sintática. Além disso, na Tabela 24 o melhor resultado é associado ao modelo com pré-processamento, estando alinhado com os resultados da Tabela 21.

Observamos para as avaliações acima, entretanto, que o conjunto de teste LX-4WAnalogiesBr só teve 35.4% (6227/17558) das perguntas avaliadas e LX-4WAnalogies, apenas 35.03% (6125/17487). As perguntas não são avaliadas quando uma das palavras da analogia não está presente no vocabulário. A baixa cobertura dos modelos suscita uma dificuldade em apontar a real acurácia de um modelo. Os resultados apontados em (HARTMANN et al.,

2017) não evidenciam a cobertura dos testes que tendem a ser omitidos pelas configurações padrão dos argumentos de avaliação (como observado no Experimento 1). (HARTMANN et al., 2017) indica que o melhor resultado geral obtido para soluções de analogias é de 46,7% com o modelo GloVe, dimensão 300 e conjunto de teste LX-4WAnalogiesBr. Todavia, sem a informação de cobertura associada, torna-se difícil a comparação entre modelos, a exemplo do respectivo *norm-stop-cbow-300* que obtém 57,49% de acurácia com 35,03% de cobertura, ambos no conjunto de teste LX-4WAnalogies. O corpus utilizado em (HARTMANN et al., 2017) dispõem de uma quantidade de tokens superior ao utilizado nesse projeto, podendo abranger mais palavras representadas.

Quando a avaliação é realizada, é utilizado por padrão um vocabulário restrito de 300000 palavras, o que tende a equalizar os modelos utilizados. Entretanto, como esses vocabulários são ordenados pela frequência de cada token no conjunto de treinamento, existe uma relação proporcional da cobertura do modelo avaliado com as palavras mais frequentes no corpus de treino. E essa relação indica que quanto melhor construída a base de dados, as palavras mais importantes serão as mais frequentes e melhor a cobertura tal como a acurácia do modelo.

Comparado o corpus de treino utilizado em (HARTMANN et al., 2017), a quantidade e variedade de dados reforça essa relação observada. Todavia, levantamos a hipótese do quanto pode ocorrer uma saturação das informações representadas. Como existe um limite de vocabulário para a avaliação de analogias, uma quantidade excessiva de dados diversos pode gerar um largo conjunto de palavras com alta frequência. E essa competição pode ser determinada por um dos tópicos semânticos que compõe o corpus treinado e que se sobrepunham aos demais tópicos. Portanto, a acurácia semântica do modelo tende a ser direcionada pelo corpus de treino e suas características. Além disso, entendemos que um conjunto de testes muito diverso tenda a ter um limite de acurácia visto esse limiar superior que os modelos possam ter quanto a sua diversidade de conteúdo.

Observamos, por outro lado, que resultados sintáticos tendam a não ser afetados por essa particularidade uma vez que são indiferentes ao tópico do corpus. Todavia, algumas linguagens podem apresentar bastante flexões e um conjunto de variações sintáticos alto, como o português. Nesses casos, entendemos que a relação é análoga ao caso semântico. O conjunto de testes, portanto, pode determinar flexões ou variações de palavras que não serão representadas no conjunto de treino.

5 CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho, buscamos explorar a técnica de Processamento de Linguagem Natural para representações vetoriais de palavras(*word embeddings*). Conforme (MIKOLOV; YIH; ZWEIG, 2013) aponta, são observados nesses espaços vetoriais regularidades linguísticas de ordens semântica e sintática. A partir dessa característica dos modelos, exploramos algumas relações presentes e como tais propriedades são utilizadas em outros trabalhos como forma de avaliação e comparação dos modelos. Como observamos uma quantidade limitada de projetos realizados para a língua portuguesa, parte do objetivo do trabalho é entender a aplicação de modelos de *word embeddings* para o respectivo idioma.

Primeiramente, observamos com base nos modelos pré-treinados disponibilizados pelo Núcleo Interinstitucional de Linguística Computacional o desempenho desses modelos para benchmarks de analogias. Usamos tais modelos também como referência para explorar as regularidades linguísticas para a língua portuguesa, analisando o comportamento do modelo conforme o processo de treinamento e de avaliação. Por fim, utilizando a ferramenta *FastText*, buscamos treinar alguns modelos com uma mesma base de dados, um dump da Wikipedia, para experimentar o processo de geração de *word embeddings*(conjunto de dados, pré-processamento e treino) e replicar o benchmark do primeiro experimento. Nessa etapa buscamos entender os resultados para modelos treinados com diferentes características e discutimos a metodologia de avaliação intrínseca.

Ao longo do trabalho observamos certas particularidades da avaliação por analogias quanto à eficiência em indicar a qualidade dos modelos e como os resultados obtidos podem ser melhor expostos através das modificações de parâmetros e adição de métricas como a cobertura do teste. (FARUQUI et al., 2016) explora a problemática da avaliação de modelos através de similaridades de palavras e aponta, por exemplo, o viés associado aos conjuntos de testes baseados em similaridade, por exemplo pares de palavras ou as analogias. Esse enviesamento ocorre porque as anotações dos testes são feitas sob uma perspectiva humana. Por exemplo, a similaridade apontada por uma instância(*banco* e *carro*) pode ser valorizada, porém, caso o modelo indique outra relação que seria válida (*banco* e *dinheiro*) o modelo será penalizado. Além disso, o estudo aponta para uma deficiência dos modelos de *word embeddings* em reproduzir o processo de divisão de aplicações de aprendizado de máquina em dividir conjuntos de dados em treino e teste. Dessa forma, o processo de ajuste do modelo para melhor acurácia tende a melhorar a eficiência apenas para o conjunto de testes, causando o que é conhecido como *overfitting*, que indica que o modelo fica especializado para um domínio específico. Essa deficiência de generalização dos modelos gera uma dificuldade de utilizar em diferentes aplicações o mesmo modelo.

Os experimentos realizados ainda indicam evidências do impacto que etapas de pré-processamento, conjunto de dados e parametrização do método de avaliação incidem sobre

o resultado dos benchmarks. Observamos como termos mal formados ou contendo caracteres e/ou pontuação são encontradas nos modelos e acabam sendo representadas sem que haja uma relevância ou utilidade direta de tais representações. Por fim, os parâmetros de avaliação detalham melhor o comportamento de modelos comparados e ressaltam a importância de uma métrica de cobertura dos testes associada à acurácia obtida. Comparado o nosso modelo com melhor acurácia, *norm-stop-cbow-300* com acurácia de 0.57, no conjunto de testes *LX-4WANALOGIES*, observamos que superamos o modelo *FastText SkipGram 300*, cuja acurácia é de 0.53 nas mesmas condições de avaliação. Todavia, a cobertura do nosso modelo atinge 35,03% do conjunto de testes, enquanto o modelo externo atinge 92,31%. Mesmo que a acurácia possa ser maior, interpretamos uma maior relevância do modelo disponibilizado pela NILC em virtude da maior representatividade do espaço vetorial.

Os experimentos realizados nesse trabalho enfrentam uma dificuldade inerente à sua proposta comparativa visto que são utilizados muitos modelos que consomem recursos computacionais. E por essa dificuldade, o escopo de alguns experimentos poderia abranger mais alternativas nos experimentos para uma avaliação aprofundada dos modelos ou o treinamento de novos. Os experimentos foram realizados em um documento *Jupyter Notebook com Python* em servidores do Departamento de Ciência da Computação, porém, ainda sim, enfrentamos limitação de espaço local para armazenar os modelos e arquivos associados.

Para trabalhos futuros, entendemos que existem extensões dos experimentos aqui realizados em virtude dos diferentes parâmetros que podem ser utilizados nas etapas descritas. A fase de pré-processamento do texto contém outras opções que são recorrentes em tarefas de PLN. Quando elaboramos essa etapa, planejamos o uso de algoritmos de redução de radicais das palavras com o objetivo de avaliar o impacto semântico em detrimento do sintático. Para tal, existem técnicas como *stemming*, que aplica regras de radiciação para sufixos determinados (por exemplo, de estudos para estudo); ou *lemmatization*, que a partir de um dicionário morfológico mapeia radicais para suas flexões. O objetivo de aplicar essas técnicas é reduzir as variações e flexões de palavras para concentrar as representações vetoriais em menos tokens do espaço. Todavia, entendemos que isso haveria de impactar a avaliação sintática do modelo, visto que essas variações costumam ser testadas nas analogias.

Na etapa de treino, observamos que o estudo aprofundado dos modelos de aprendizado de máquina para geração de *word embeddings* disponíveis, como (MIKOLOV et al., 2013), (BOJANOWSKI et al., 2017) ou (PENNINGTON; SOCHER; MANNING, 2014) e o ajuste dos parâmetros relacionados poderia ser explorada para também avaliar os potenciais resultados. A ferramenta do *FastText*, por exemplo, permite a alteração do parâmetro referente à janela de informação em sub-palavras. Como utilizamos o respectivo algoritmo para treino, entendemos que há também a oportunidade de medir os resulta-

dos associados a essa parametrização em virtude da diversidade morfológica da língua portuguesa.

Para a avaliação e comparação dos resultados dos modelos pré-treinados, o parâmetro de vocabulário restrito poderia ser utilizado para avaliar, para modelos de corpus de treino diferentes, a sensibilidade da acurácia. O ajuste do parâmetro também pode ser relevante para compor a avaliação baseada na cobertura dos modelos. Outro fator importante na avaliação de analogias é a alternativa do método *3CosMul*(LEVY; GOLDBERG, 2014). De forma simplificada, a propriedade multiplicativa do método busca valorizar melhor as relações das palavras e distinguir melhor as similaridades, evitando a predominância de palavras na questão. Uma sugestão de trabalho futuro é reproduzir as propostas de (LEVY; GOLDBERG, 2014) e avaliar para conjuntos de dados em português os resultados obtidos através do método *3CosMul*.

Assim como (BAKAROV, 2018) lista conjuntos de teste para avaliação de modelos, notamos a oportunidade de colaboração tanto para métodos de avaliação e benchmark como para modelos pré-treinados. Primeiramente, as ferramentas utilizadas para criação de modelos disponibilizam formas de prover mais dados aos mesmos, com o intuito de agregar e aprimorar as representações existentes com mais contextos. Os conjuntos de testes, por sua vez, podem servir como referência para novos conjuntos que cubram diferentes relações. Além disso, tendo em vista o potencial semântico desses modelos, entendemos que formular conjuntos de testes mais amplos semanticamente pode demonstrar melhor a acurácia dos modelos para tópicos diversificados.

(MIKOLOV et al., 2018) indica duas propriedades para os modelos treinados que poderiam ser explorados para a língua portuguesa. A primeira é a composição de palavras, formando estruturas maiores das representações para o nível de expressões, sentenças ou documentos. Em nossos experimentos, observa-se que não podemos utilizar palavras compostas uma vez que o treino é feito a nivel de palavras, ou sub-palavras no caso do *FastText*. Portanto, para palavras como *eua* ou *obama*, acreditamos que utilizar a representação de *Estados Unidos* ou *Barack Obama* poderia alterar os resultados de similaridade. A outra propriedade é observada nos modelos Skipgram e indica resultados promissores para operações de adição entre vetores. Conforme os resultados apontado, seria possível realizar a consulta $\overrightarrow{Germany} + \overrightarrow{capital} = \overrightarrow{Berlim}$. Todavia, quando experimentamos durante a fase exploratória a equivalente operação em português ou uma consulta análoga ($\overrightarrow{Obama} + \overrightarrow{esposa}$), o resultado não foi satisfatório. Para trabalhos futuros, o estudo dessa propriedade é a oportunidade de explorar semanticamente as informações contidas no corpus de treino a partir de uma consulta com duas palavras, sem a necessidade de uma analogia completa.

Por fim, a sugestão para próximos trabalhos é também utilizar tarefas associadas a PLN para avaliar os modelos treinados. A avaliação extrínseca, como sugere (FARUQUI et al., 2016), poderia evitar problemas inerentes aos modelos de word embeddings e me-

lhor representar a qualidade dos modelos visto o ganho que tarefas adquirem ao utilizar tais representações de palavras. Além disso, medir a correlação entre as duas formas de avaliação pode indicar a representatividade das regularidades linguísticas absorvidas pelas representações vetoriais das palavras em aplicações práticas de Processamento de Linguagem Natural.

REFERÊNCIAS

ALPAYDIN, E. **Machine learning: the new AI**. [S.l.]: MIT press, 2016.

BAKAROV, A. A survey of word embeddings evaluation methods. **arXiv preprint arXiv:1801.09536**, 2018.

BENGIO, Y. et al. A neural probabilistic language model. **J. Mach. Learn. Res.**, JMLR.org, v. 3, n. null, p. 1137–1155, mar. 2003. ISSN 1532-4435.

BOJANOWSKI, P. et al. Enriching word vectors with subword information. **Transactions of the Association for Computational Linguistics**, v. 5, p. 135–146, 2017. Disponível em: <https://www.aclweb.org/anthology/Q17-1010>.

EL-AMINE, M. NLP Research Lab Part 1: Distributed Representations. **Medium**, dez. 2017. Disponível em: <https://medium.com/district-data-labs/nlp-research-lab-part-1-distributed-representations-b7296b522d38>. Acesso em: 24 set. 2020.

FARUQUI, M. et al. Problems with evaluation of word embeddings using word similarity tasks. **arXiv preprint arXiv:1605.02276**, 2016.

GOOGLE. Embeddings | Machine Learning Crash Course. **Google Developers**, 2020. Disponível em: <https://developers.google.com/machine-learning/crash-course/embeddings/video-lecture?hl=pt-br>. Acesso em: 24 set. 2020.

GUTHRIE, D. et al. A closer look at skip-gram modelling. In: **Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)**. Genoa, Italy: European Language Resources Association (ELRA), 2006. Disponível em: <http://www.lrec-conf.org/proceedings/lrec2006/pdf/357_pdf.pdf>.

HARTMANN, N. et al. Portuguese word embeddings: Evaluating on word analogies and natural language tasks. In: **Proceedings of the 11th Brazilian Symposium in Information and Human Language Technology**. Uberlândia, Brazil: Sociedade Brasileira de Computação, 2017. p. 122–131. Disponível em: <https://www.aclweb.org/anthology/W17-6615>.

HINTON, G. E.; MCCLELLAND, J. L.; RUMELHART, D. E. Distributed representations. In: _____. **Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations**. Cambridge, MA, USA: MIT Press, 1986. p. 77–109. ISBN 026268053X.

IACOBACCI, I.; PILEHVAR, M. T.; NAVIGLI, R. Embeddings for word sense disambiguation: An evaluation study. In: **Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**. Berlin, Germany: Association for Computational Linguistics, 2016. p. 897–907. Disponível em: <https://www.aclweb.org/anthology/P16-1085>.

JAMES, G. et al. **An introduction to statistical learning**. [S.l.]: Springer, 2013. v. 112.

LEVY, O.; GOLDBERG, Y. Linguistic regularities in sparse and explicit word representations. In: **Proceedings of the eighteenth conference on computational natural language learning**. [S.l.: s.n.], 2014. p. 171–180.

LING, W. et al. Two/too simple adaptations of word2vec for syntax problems. In: **Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**. [S.l.: s.n.], 2015. p. 1299–1304.

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

MIKOLOV, T. et al. Advances in pre-training distributed word representations. In: **Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)**. [S.l.: s.n.], 2018.

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**. Atlanta, Georgia: Association for Computational Linguistics, 2013. p. 746–751. Disponível em: <https://www.aclweb.org/anthology/N13-1090>.

MITCHIE, D.; SPIEGELHALTER, D.; TAYLOR, C. **Machine learning, neural and statistical classification, 1994**. Ellis Horwood, New York). Google Scholar, 1994. Disponível em: <http://www1.maths.leeds.ac.uk/~charles/statlog>.

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

PYTHON module · fastText. s.d. Publisher: FACEBOOK. Disponível em: <https://fasttext.cc/index.html>. Acesso em: 23 out. 2020.

RODRIGUES, J. et al. Lx-dsemvectors: Distributional semantics models for portuguese. In: SILVA, J. et al. (Ed.). **Computational Processing of the Portuguese Language**. Cham: Springer International Publishing, 2016. p. 259–270. ISBN 978-3-319-41552-9.

SCHROFF, F.; KALENICHENKO, D.; PHILBIN, J. Facenet: A unified embedding for face recognition and clustering. **2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, IEEE, Jun 2015. Disponível em: <http://dx.doi.org/10.1109/CVPR.2015.7298682>.

APÊNDICES

APÊNDICE A – DOCUMENTAÇÃO DOS EXPERIMENTOS

O trabalho foi realizado na ferramenta *Notebook Jupyter* que é uma interface *python* para organização de código e texto de forma integrada, e disponibilizada em um dos servidores do Departamento de Ciência da Computação da UFRJ. O arquivo de referência base para os experimentos deste trabalho estão disponíveis, até o momento de publicação do mesmo, em https://github.com/bferrazAC/avaliacao-por-analogias.

ANEXOS

ANEXO A – PARÂMETROS DO TREINO DE *WORD EMBEDDINGS*

O método para geração dos Word Embeddings tem definido os seguintes parâmetros.

Tabela 25 – Acurácia Sintatica

Parâmetro	Descrição	Valor Padrão
input	caminho arquivo de treinamento	Parâmetro Obrigatório
model	modelo não supervisionado fasttext	[skipgram]
lr	taxa de aprendizado	[0.05]
dim	tamanho dos vetores de palavras	[100]
ws	tamanho da janela de context	[5]
epoch	numero de epocas	[5]
minCount	numero minino de ocorrências	[5]
minn	tamanho mínimo de ngram de caracteres	[3]
maxn	tamanho máximo de ngram de caracteres	[6]
neg	numero de amostras negativas	[5]
wordNgrams	tamanho máximo de ngram de palavras	[1]
loss	função de perda	[ns]
bucket	número de *buckets*	[2000000]
thread	numero de *threads*	[number of cpus]
lrUpdateRate	taxa de atualização para a taxa de aprendizado	[100]
t	limiar de amostras	[0.0001]
verbose	verboso	[2]

Traduzido de (PYTHON. . . , s.d)