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

VICTOR GARRITANO NORONHA

RELACIONANDO GEOMETRICAMENTE TWEETS E NOTÍCIAS UTILIZANDO A WORD MOVER'S DISTANCE

RIO DE JANEIRO
2018

VICTOR GARRITANO NORONHA

RELACIONANDO GEOMETRICAMENTE TWEETS E NOTÍCIAS UTILIZANDO A WORD MOVER'S DISTANCE

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 Antonio Recio da Paixão
Co-orientador: M. Sc. Felipe Fink Grael

RIO DE JANEIRO
2018

CIP - Catalogação na Publicação

N852r

Noronha, Victor Garritano
Relacionando geometricamente tweets e notícias utilizando a Word Mover's Distance / Victor Garritano Noronha. -- Rio de Janeiro, 2019.
53 f.

Orientador: João Antonio Recio da Paixão.
Coorientador: Felipe Fink Grael.
Trabalho de conclusão de curso (graduação) - Universidade Federal do Rio de Janeiro, Instituto de Matemática, Bacharel em Ciência da Computação, 2019.

1. recuperação de informação. 2. word embeddings. 3. similaridade entre documentos. 4. alinhamento de espaços vetoriais. I. Paixão, João Antonio Recio da, orient. II. Grael, Felipe Fink, coorient. III. 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.

VICTOR GARRITANO NORONHA

RELACIONANDO GEOMETRICAMENTE TWEETS E NOTÍCIAS UTILIZANDO A WORD MOVER'S DISTANCE

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 23 de janeiro de 2019

BANCA EXAMINADORA:

Prof. João Antonio Recio da Paixão
D.Sc. - PUC

Felipe Fink Grael
M.Sc. - COPPE/UFRJ

Prof. João Carlos Pereira da Silva
D.Sc. - COPPE/UFRJ

Prof. Felipe Maia Galvão França
Ph.D. - Imperial College London

Fernando Guimarães Ferreira
D.Sc. - COPPE/UFRJ

AGRADECIMENTOS

Agradeço a Jesus, meu Senhor e Salvador, por me dar forças, ânimo e sabedoria durante todos os momentos, e por me fazer chegar até a conclusão dessa etapa de minha vida. A Ele seja toda honra e glória.

Agradeço a minha família, em especial aos meus pais, Nadia e Adilson e a minha irmã Vivian, por me apoiarem e me incentivarem durante toda a graduação, sempre me motivando a não desistir e seguir em frente.

Agradeço aos meus amigos, em especial os que estiveram comigo durante a graduação, por todos os momentos vividos juntos e aprendizados compartilhados, na vida dentro e fora da sala de aula.

Agradeço aos professores do Departamento de Ciência da Computação da Universidade Federal do Rio de Janeiro por tudo que foi ensinado e aprendido durante esses anos em que convivemos juntos.

Deixo meu agradecimento especial por todas as palavras de conselho e incentivo aos professores João Carlos, Juliana e João Paixão (que atuou como meu orientador no projeto final). Sou muito grato por toda a dedicação de vocês como professores e amigos, pois sem dúvidas vocês foram fundamentais para que hoje eu me tornasse um Cientista da Computação.

Registro também um agradecimento final à Twist, empresa da qual faço parte desde 2018. Obrigado a todos os colaboradores e sócios, em especial ao Felipe Grael, meu coorientador, por todos os momentos juntos, toda a ajuda, todo o incentivo, todos os conselhos e desafios superados juntos.

RESUMO

A maneira de consumir e produzir notícias mudou muito ao longo dos anos. Os portais de notícias perderam sua exclusividade como produtores de informação devido à grande popularização das redes sociais. Faz-se necessário, portanto, conseguir acompanhar a repercussão das informações que circulam por essas duas fontes. No entanto, cada um desses meios possui características particulares a respeito do estilo de escrita: textos jornalísticos apresentam uma linguagem formal, ao passo que postagens de redes sociais fazem constante uso de gírias e neologismos, e também podem apresentar erros ortográficos e gramaticais com mais frequência. Além disso, fatos frequentemente possuem repercussão em vários idiomas. Por isso, este trabalho explora técnicas de recuperação de informações multilíngue, baseadas em trabalhos anteriores, para melhor aproveitar as características de cada texto. As palavras de cada documento são representadas por *word embeddings*, o que permite que informações semânticas sejam levadas em consideração no cálculo das similaridades. Realiza-se então o alinhamento desses espaços vetoriais, para estabelecer relações de significados entre os *embeddings* de meios e idiomas diferentes. Com isso é possível representar documentos como uma nuvem de pontos em um espaço vetorial comum, e utilizar uma técnica baseada em transporte ótimo de massa para o cálculo da similaridade. Essa abordagem traz ganhos de 15% na precisão em comparação com trabalhos anteriores.

Palavras-chave: recuperação de informação. *word embeddings*. similaridade entre documentos. alinhamento de espaços vetoriais.

ABSTRACT

In recent years, the way people produce and consume content has changed. News portals have lost their exclusivity as information producers due to the increase in popularity of social networks. Therefore, one must be able to monitor the impact of information spread through these two content sources. However, each of these media has particular characteristics regarding the style of writing: newspaper texts exhibit a formal language, while social network posts make constant use of slangs and neologisms, and can also present orthographical and grammatical errors more frequently. Besides, facts are frequently spread in many languages. Therefore, this work investigates multilingual information retrieval techniques, based on previous works, to better take advantage of the characteristics of each text. The words in each document are represented by word embeddings, which allows the semantic information to be taken into account in the calculation of similarities. The vector space alignment is performed in order to establish semantic relationships among the embeddings of different languages. This way it's possible to represent documents as a point cloud in a common vector space, and use a technique based on optimal transport for the similarity calculation. This approach exhibits a precision improvement of 15% in relation to previous works.

Keywords: information retrieval. word embeddings. document similarity. vector space alignment.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

En	Inglês
Fr	Francês
EMD	Earth Mover's Distance
CBOW	Continuous Bag-of-Words
SVD	Decomposição em valores singulares
WMD	Word Mover's Distance
TF	Term Frequency
IDF	Inverse Document Frequency
KL	Kullback-Leibler
JS	Jensen-Shannon

SUMÁRIO

1 INTRODUÇÃO

A maneira de consumir e produzir conteúdo ao longo dos últimos anos mudou radicalmente (MARTINEZ-ALVAREZ et al., 2016). Atualmente, além da tradicional veiculação de notícias por meio de jornais e revistas, existe um grande consumo de informação através de portais eletrônicos publicamente disponíveis na internet. Porém esse consumo não se limita apenas a esses portais: a popularização das redes sociais gerou uma nova maneira de consumirmos conteúdo.

Entretanto, não foi apenas a forma de consumir notícias que mudou: as próprias redes sociais se tornaram um novo polo de produção de informação. Através do Twitter, Facebook e Instagram, por exemplo, qualquer usuário pode reportar algum acontecimento que está ocorrendo em sua localidade em tempo real; esses mesmos usuários podem reagir e participar de discussões sobre diversos assuntos pelo mundo.

1.1 MOTIVAÇÃO

É possível acompanhar a evolução de acontecimentos mundiais através da repercussão gerada na redes sociais. Estudos recentes analisaram como essas redes, em especial o Twitter, podem ser utilizadas como uma valiosa fonte de novos acontecimentos e opinião pública. Entre esses trabalhos, podemos citar uma análise dos conteúdos presentes em tweets como substituição para pesquisas de opinião (BALASUBRAMANYAN; ROUTLEDGE; SMITH, 2010); uma tentativa de prever as movimentações do mercado financeiro através do Twitter (BOLLEN; MAO; ZENG, 2011) e ainda um sistema de detecção e alerta de terremotos baseado no monitoramento de tweets (SAKAKI; OKAZAKI; MATSUO, 2010).

Outro estudo que evidencia a eficácia dos tweets no que diz respeito a capturar e influenciar a opinião pública foi apresentado em ASUR; HUBERMAN (2010): através do monitoramento de usuários interagindo e discutindo sobre um determinado filme, os autores conseguiram analisar como os sentimentos sobre uma obra cinematográfica são criados, como as opiniões são geradas e como elas se propagam através das redes sociais, conseguindo prever assim o sucesso de bilheteria que aquele filme teria. Entre as conclusões obtidas, ficou constatado que o sentimentos por trás dos primeiros tweets são fatores determinantes para um sucesso ou fracasso.

Assim, fica claro como os acontecimentos mundiais e as redes sociais estão interligados, e como eles tem a capacidade de influenciar um ao outro. Isso não é diferente para o âmbito da comunicação e do jornalismo: em CHONG; CHUA (2013) e ALSAEDI; BURNAP; RANA (2016), foram estudadas maneiras de gerar relatórios dos principais acontecimentos mundiais através do resumo da informação presente nas mensagens do

Twitter. Aqui, podemos observar como as redes sociais não apenas consomem informação, mas efetivamente produzem esse conteúdo que será consumido pelas pessoas no mundo.

Outro caso recente foi o *Reuters Tracer* (LIU et al., 2018), um sistema que, segundo os autores, seria capaz de automatizar totalmente a produção de notícias com base em dados do Twitter, sendo capaz de realizar todo o processo jornalístico, desde a detecção de eventos relevantes em tempo real até a disseminação das notícias geradas, sem nenhuma intervenção humana.

1.2 OBJETIVO

Com toda essa influência que as redes sociais e os portais de notícias exercem um sobre o outro, é evidente que esses meios de disseminação de conteúdo estão bastante interligados, sendo de crucial importância a tarefa de acompanhar a repercussão de um determinado evento por essas diferentes fontes, evidenciando a forma como as informações são propagadas nesses meios de comunicação naturalmente distintos, uma vez que os portais de notícias possuem uma linguagem formal e padronizada, enquanto os tweets e textos de redes sociais em geral apresentam uma forma de escrita totalmente livre, com abreviações, neologismos e palavras escritas de maneira incorreta. Nesse trabalho, estudaremos um método para detectar relações semânticas entre os conteúdos presentes no Twitter e em portais de notícias, encontrando os tweets que tratam do mesmo assunto de uma notícia, e vice-versa.

1.3 TRABALHOS RELACIONADOS

A tentativa de estabelecer relações entre as informações disseminadas pelo Twitter e por portais de notícias é uma tarefa já explorada anteriormente. Em ZHAO et al. (2011), os autores utilizaram técnicas de modelagem de tópicos para comparar os assuntos que estavam sendo abordados no Twitter e no *New York Times*, um jornal americano. Através desses estudos, eles comprovaram que os assuntos mais falados no portal de notícia e na rede social são bastante semelhantes, e que a rede social auxilia bastante na disseminação das principais notícias.

Em GUO et al. (2013) os autores procuraram encontrar notícias semelhantes a um determinado tweet, como forma de aumentar a base de dados para tarefas de Processamento de Linguagem Natural que lidam com textos de redes sociais. Utilizando as *hashtags* presentes no tweets e as entidades nomeadas presentes nas notícias, os autores foram capazes de estabelecer correlações entre as diferentes fontes de informação, e agregar mais conhecimento os textos da rede social.

Em TSAGKIAS; RIJKE; WEERKAMP (2011) foi apresentada uma abordagem para uma tarefa semelhante ao problema que é o objeto de estudo deste trabalho. Nesse artigo, o objetivo é encontrar, dada uma notícia, os tweets que referenciavam implicitamente

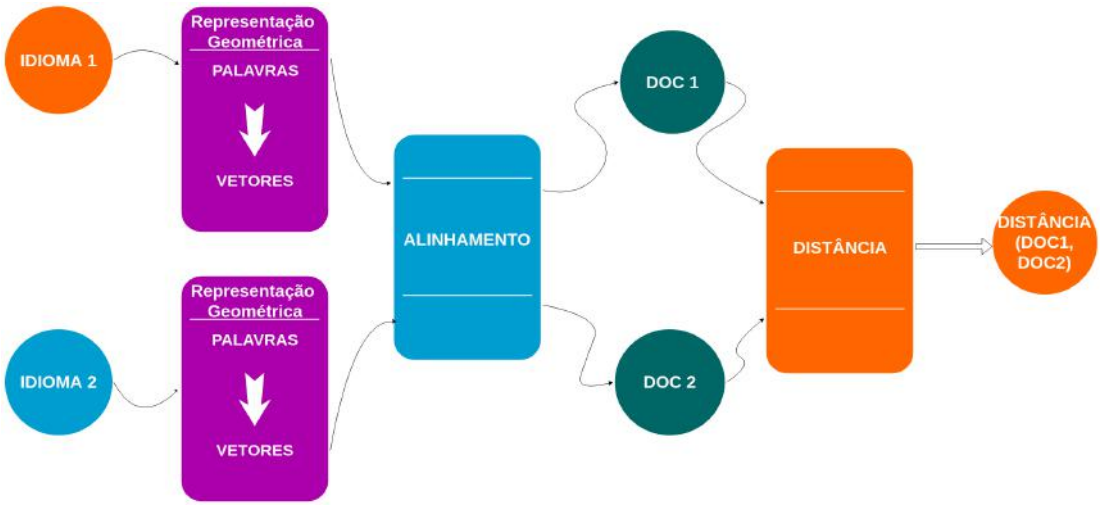
aquele texto jornalístico, estabelecendo, dessa forma, uma ligação entre as notícias e sua repercussão nas redes sociais. Um fato bem interessante a ser destacado é a preocupação dos autores em fazer a ligação entre os vocabulários utilizados nessas duas fontes de informação.

Outro trabalho que segue a linha de fazer a ligação entre a linguagem dos tweets e das notícias foi apresentado em MOGADALA; JUNG; RETTINGER (2017). Os autores propuseram uma transformação não-linear como uma forma de aproximar os vocabulários um do outro através de representações vetoriais, os *word embeddings*, e assim facilitar a identificação de similaridades entre tweets e notícias.

1.4 FORMULAÇÃO DA ABORDAGEM

Nesse trabalho, nós apresentaremos uma abordagem para a tarefa de encontrar tweets e notícias que tratem de um mesmo assunto, sendo, portanto, semanticamente parecidos. Para esse problema, utilizaremos a abordagem proposta em BALIKAS et al. (2018), que tem por objetivo estabelecer um grau de semelhança semântica entre documentos de universos diferentes. Essa formulação, originalmente aplicado a tarefa de relacionar páginas da Wikipedia em inglês e em francês, foi adaptado para o contexto de relacionar tweets e notícias. A Figura 1 destaca os principais componentes da mesma.

Figura 1 – Blocos Constituintes da Abordagem



A primeira etapa é a representação de tweets e notícias através da utilização de espaços vetoriais de palavras; em seguida, entra o componente responsável por aproximar o vocabulário desses dois universos, que é o alinhamento de espaços vetoriais. Por fim, com as representações vetoriais alinhadas, podemos efetivamente comparar a similaridade semântica entre um tweet e uma notícia recorrendo a uma métrica para medir distância, que leva em consideração as semelhanças de significado entre as representações vetoriais das palavras que compõem cada um dos documentos.

1.5 CONJUNTO DE DADOS

Aplicaremos a abordagem sobre uma base de dados apresentada em SUAREZ et al. (2018), que estabelece relações de relevância entre tweets e notícias, publicamente disponível. A base de dados é composta por 99 notícias e 5865 tweets onde, para cada notícia, foi estabelecido um grau de relevância entre a mesma e alguns tweets selecionados aleatoriamente, segundo os autores. Assim, cada notícia possui, em média, um grau de relevância semântica estabelecido com 61 tweets.

Esse grau de relevância é binário: caso um tweet trate do mesmo assunto de uma notícia, a relevância entre ambos é assinalada como `1`, caso contrário, ela recebe o valor `0`. Dessa forma, dada uma notícia, a tarefa é conseguir recuperar os tweets que são de fato relevantes para ela.

No artigo que forneceu a base de dados, estão presentes alguns resultados que fornecem um ponto de partida para comparações de desempenho na tarefa. Esses resultados foram obtidos através de diversas consultas a um servidor de buscas, utilizando fragmentos das notícias como corpo da consulta. Esse servidor retornaria os tweets mais semelhantes àquela notícia.

Para este trabalho, algumas notícias foram removidas da base de dados, uma vez que elas não possuíam um número suficiente de tweets relacionados como relevantes. No artigo original, não fica claro se essa medida foi adotada ou não, uma vez que a maior contribuição do artigo foi a base de dados anotada, porém julgamos que essa filtragem faz-se necessária, por conta da medida de desempenho que foi utilizada em SUAREZ et al. (2018) e por partirmos do pressuposto que a medida escolhida é de fato adequada para o conjunto de dados sobre o qual os resultados originais foram reportados.

Esse trabalho é organizado como segue: nos capítulos seguintes, discutiremos cada etapa de nossa abordagem: no Capítulo 2 discutiremos sobre as representações vetoriais de palavras, os *word embeddings*; no Capítulo 3 trataremos sobre o alinhamento dos espaços vetoriais dos tweets e das notícias; no Capítulo 4 apresentaremos a distância que utilizaremos como indicador do grau de similaridade entre um tweet e uma notícia. A seguir, no Capítulo 5, apresentaremos a metodologia adotada; no Capítulo 6, mostraremos e discutiremos os resultados obtidos e finalmente, no Capítulo 7, exporemos algumas conclusões.

2 REPRESENTAÇÕES VETORIAIS DE PALAVRAS

Iniciaremos a discussão da primeira etapa de nossa abordagem, que são as representações vetoriais de palavras. De um modo geral, essa primeira etapa realiza uma mudança de representação nos documentos, levando de palavras para uma forma mais adequada para a manipulação no computador.

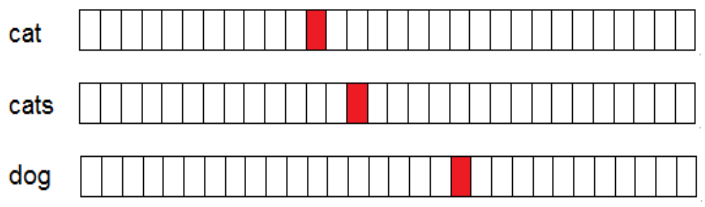
A principal vantagem de utilizarmos essa mudança de representação é que as strings, enquanto cadeia de caracteres, pouco nos dizem a respeito do significado que as palavras carregam. Utilizando apenas *strings*, ficamos limitados às relações semânticas e sintáticas que podemos capturar, como sufixos (feliz → felizmente) e mudanças entre singular e plural (gato → gatos), por exemplo. Considerando as cadeias de caracteres, ficamos limitados a capturar apenas relações de significado que compartilham caracteres entre palavras.

Isso não é o suficiente, uma vez que, como é de conhecimento geral, palavras com semântica semelhante não necessariamente compartilham os mesmos caracteres ou fragmentos em comum, como andar e caminhar, boiar e flutuar, por exemplo, ou ainda relações mais complexas que sinônimos, como capitais (Grécia → Atenas), ou parentescos (genro → nora). Esses são só alguns exemplos que ilustram o fato de que é necessário uma representação que consiga representar melhor essa grande quantidade de relações sintáticas e semânticas que existem entre as mais diversas palavras. Essa maneira mais adequada de representar os dados são os *embeddings*.

Podemos entender os *embeddings* em geral como um mapeamento de algum objeto, seja uma imagem, um texto, um sinal de áudio, para objetos matemáticos, como números, vetores e matrizes. Podemos representar uma imagem como uma matriz de intensidades de cada pixel, ou ainda um sinal de áudio com um espectro de frequências, por exemplo. Nesse trabalho, focaremos nas representações específicas para texto.

Existem algumas formas para obtermos essa representação. Uma forma possível seria criar um vetor com tamanho do vocabulário da base de dados, ou seja, com o número de palavras únicas presentes naquela base e associar para cada posição desse vetor a ocorrência de uma palavra única no texto. Essa abordagem é conhecida como *one-hot encoding*, e está exemplificada na Figura 2.

Figura 2 – Exemplo de one-hot encoding. Extraído de SHWARTZ (2006)



Porém essa representação ainda não nos fornece as relações semânticas que precisamos. Assim, iremos recorrer a uma terceira forma de representar palavras e textos, que pode nos fornece as relações que estamos procurando, os *word embeddings*.

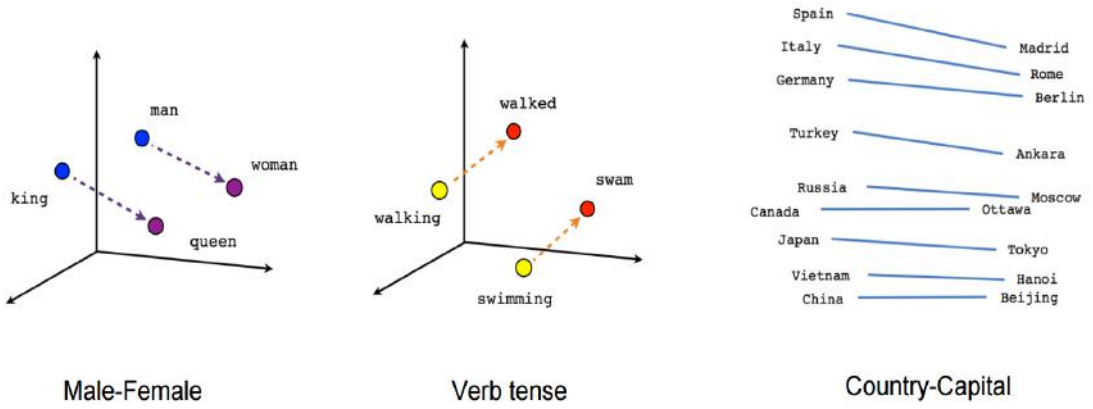
2.1 WORD EMBEDDINGS

Esse tipo de representação pode ser encarado como um espaço vetorial, de dimensão fixa, onde espera-se que palavras com significados semelhantes estejam em posições próximas nesse espaço.

A representação de palavras como vetores é uma abordagem relativamente antiga, sendo introduzida por BENGIO; DUCHARME; VINCENT (2003), porém apenas a partir de 2013 esse tipo de representação começou a ganhar popularidade e prestígio, graças a MIKOLOV et al. (2013a).

Nesse trabalho, foram propostas algumas simplificações sobre o modelo original, permitindo a escalabilidade da técnica de representação para grandes volumes de dados, além de comprovar a eficiência da representação de palavras utilizando espaços vetoriais, no que diz respeito a capturar relações sintáticas e semânticas entre as palavras, como as que estão ilustradas na Figura 3.

Figura 3 – Exemplos de relações capturadas por Word Embeddings. Extraído de TENSORFLOW (2018)



Uma característica observada nesses espaços vetoriais, e que foi de certa forma surpreendente para os próprios autores, é que podemos realizar analogias entre palavras, realizando operações matemáticas simples, como soma e subtração de vetores.

Vamos às seguintes definições: seja `vetor(x)` o ponto do espaço vetorial associado à palavra `x`. Agora, sejam dois vetores a e b, e a_i e b_i as i-ésimas coordenadas dos vetores a e b, respectivamente. Temos que a similaridade cosseno entre esses vetores é expressa por:

$$\frac{\sum_i a_i \cdot b_i}{\sqrt{\sum_i a_i^2} \cdot \sqrt{\sum_i b_i^2}} \tag{2.1}$$

Se calcularmos por exemplo, como apresentado em (MIKOLOV et al., 2013a), `vetor(biggest) - vetor(big) + vetor(small)` e determinarmos o vetor mais próximo do vetor resultante, em termos de similaridade cosseno, por exemplo, obteremos o vetor correspondente a `smallest`; se realizarmos a operação `vetor(Messi) - vetor(midfielder) + vetor(scientist)`, obteremos um vetor cujo vizinho mais próximo será o vetor correspondente a `Einstein`. A Tabela 1 ilustra mais algumas relações semânticas que podemos capturar utilizando essas simples operações algébricas.

Tabela 1 – Exemplos de analogias entre palavras

Analogia	**Resultado**
France - Paris + Italy	Rome
Sarkozy - France + Berlusconi	Italy
Japan - sushi + Gernany	bratwurst
Berlusconi - Silvio + Obama	Barack
Microsoft - Windows + Google	Android

Na coluna esquerda, realizamos as operações matemáticas com os vetores correspondentes às palavras indicadas. Na coluna direita, consta a palavra associada ao vetor mais próximo ao vetor resultante da operação. Adaptada de (MIKOLOV et al., 2013a)

Existem diversos problemas onde é possível tirar proveito desses espaços vetoriais e suas relações semânticas, com o objetivo de melhorar o desempenho dos métodos empregados para resolvê-las. Como exemplos de tarefas em que o desempenho foi melhorado graças aos *word embeddings*, podemos citar o reconhecimento de entidades nomeadas (LAMPLE et al., 2016), que basicamente é a tarefa de identificar nomes relevantes, como pessoas, cidades e organizações em um texto; a Part-of-Speech Tagging (PLANK; SØGAARD; GOLDBERG, 2016), que é essencialmente a tarefa de classificar as classes gramaticais das palavras em um texto, identificando se as mesmas atuam como nomes, pronomes, verbos, advérbios, etc; a análise de dependências, ou Dependency Parsing (YU; VU, 2017), onde procura-se entender as relações semânticas entre as palavras; e a modelagem de linguagem (HOWARD; RUDER, 2018), que basicamente atribui probabilidades a frases, definindo a chance de uma frase ser um frase válida, do ponto de vista sintático e semântico.

2.2 CONSTRUÇÃO DE ESPAÇOS VETORIAIS

Uma vez apresentadas e exemplificadas as capacidades e o poder semântico dos *word embeddings*, resta discutirmos como esses espaços são construídos. Apresentaremos bre-

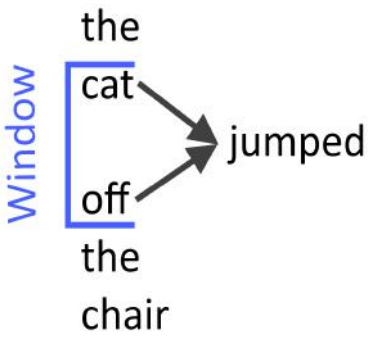
vemente a seguir alguns dos métodos mais utilizados para gerar esses espaços vetoriais de palavras: o Word2Vec (MIKOLOV et al., 2013a) e o FastText (BOJANOWSKI et al., 2017).

Os métodos em geral se baseiam na chamada hipótese distribuída (do inglês *distributional hypothesis*), que foi apresentada em (HARRIS, 1954) e atesta que palavras que aparecem em contextos parecidos tendem a ter significados parecidos. Uma outra interpretação para essa frase é dizer que palavras que são antecedidas e sucedidas por outras palavras em comum, tendem a ter significado semelhante.

2.2.1 Word2Vec

O primeiro método a se popularizar para a geração de *word embeddings* foi o Word2Vec. O algoritmo possui duas versões, que variam conforme a saída esperada de cada uma delas. A primeira versão, o CBOW, ou *Continuous Bag-of-Words* tem a seguinte proposta: gerar uma palavra central, dado o seu contexto. Em outras palavras, gerar uma palavra central, dadas n palavras a frente e n palavras atrás. Por exemplo, na frase `the cat jumped off the chair`, se considerarmos como palavra central o termo `jumped`, objetivo é obter `jumped`, dado o contexto, que pode ter seu tamanho variável (`cat` e `off` nesse caso com $n = 1$), como exemplificado na Figura 4:

Figura 4 – Continuous Bag-of-words. Extraído de SOLUTIONS (2016)

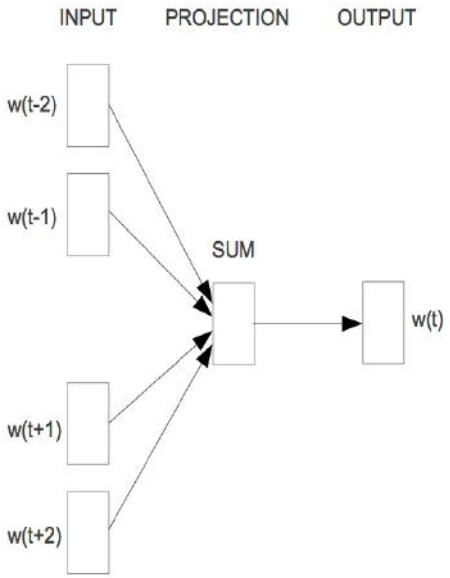


Ou seja, o objetivo é maximizar a probabilidade da palavra `jumped` ser retornada pelo método, dado que as palavras `cat` e `off` apareceram na janela de contexto. Na Figura 5 utilizaremos uma ilustração um pouco mais abstrata para introduzir a formalização, por meio de uma função, desse objetivo citado anteriormente.

Seja w_t a palavra central do texto na posição t , sendo M a última posição possível; seja w_i a palavra que se encontra na posição i; seja ainda $p(w_i|w_j)$ a probabilidade da palavra w_i ser retornada, dado que w_j apareceu. A função objetivo que queremos maximizar é dada por:

$$\frac{1}{M}\sum_{t=1}^{M}\log p(w_t|w_{t-n},\cdots,w_{t-1},w_{t+1},\cdots,w_{t+n}) \tag{2.2}$$

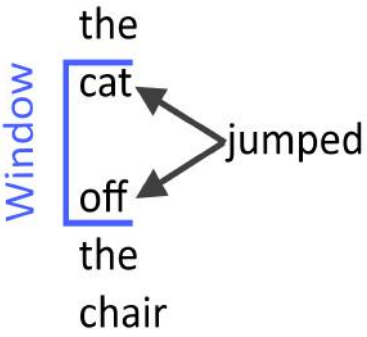
Figura 5 – Representação Genérica do CBOW. Extraído de MIKOLOV et al. (2013a)



Ou seja, queremos maximizar a probabilidade de, dadas as palavras do contexto daquela palavra central, gerarmos a palavra central propriamente dita.

A segunda versão, a Skip-Gram, por sua vez, tenta prever o contexto em si, dada a palavra central. Retomando o exemplo anterior, o objetivo é prever as palavras do contexto `cat` e `off`, dada a palavra central `jumped`, como ilustrado na Figura 6. Assim como fizemos no caso do CBOW, introduziremos uma ilustração do algoritmo, na Figura 7.

Figura 6 – Skip-Gram. Extraído de SOLUTIONS (2016)

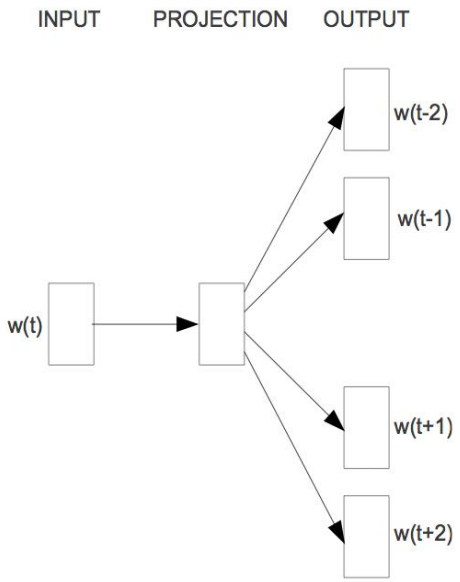


Seja n a quantidade de palavras a frente e atrás utilizadas como contexto. A função objetivo a ser maximizada, nesse caso, será:

$$\frac{1}{M}\sum_{t=1}^{M}\sum_{-n\leq j\leq n, j\neq 0}\log p(w_{t+j}|w_t) \tag{2.3}$$

O objetivo é maximizar a probabilidade de, dada uma certa palavra central t, gerar as palavras do contexto $w_{t-n},\ldots,w_{t-1},w_{t+1},\ldots,w_{t+n}$.

Figura 7 – Representação Genérica do Skip-Gram. Extraído de MIKOLOV et al. (2013a)



Para cada uma das versões, as probabilidades podem ser calculadas utilizando uma função conhecida como *softmax*. No Skip-Gram, a probabilidade de uma palavra de contexto w_c ser gerada, dado que a palavra central w_t apareceu, é:

$$p(w_c|w_t) = \frac{e^{\vec{w_c}^T \vec{w_t}}}{\sum_{w_k \in \mathbf{W}} e^{\vec{w_k}^T \vec{w_t}}} \tag{2.4}$$

onde $\mathbf{W}$ é o conjunto de todas as palavras presentes no texto e $\vec{w_c}^T \vec{w_t}$ representa o produto interno entre os vetores $\vec{w_c}$ e $\vec{w_t}$, que são as representações das palavras w_c e w_t, respectivamente. Podemos pensar nesse produto interno como um grau de similaridade entre os vetores que representam essas palavras.

Utilizando o método do gradiente descendente, precisamos calcular o gradiente de nossa função objetivo (2.4); porém, como informado pelo autores em (MIKOLOV et al., 2013b), calcular o gradiente $\nabla p(w_c|w_t)$ (que é a informação necessária para a maximização) exatamente é uma operação computacionalmente custosa. Logo se fazem necessárias alternativas para aproximar o cálculo dessas probabilidades.

Uma opção mais computacionalmente eficiente para aproximar o *softmax* é, por exemplo, o *hierarchical softmax* (MORIN; BENGIO, 2005), que utiliza uma estrutura de árvore binária para reduzir de $\mathbf{W}$ para $\log_2(\mathbf{W})$ o número de palavras que precisamos avaliar na hora de obter a distribuição de probabilidade do *softmax* e calcular seu gradiente.

2.2.2 FastText

Outro método existente para a geração de *embeddings* é o FastText, introduzido em BOJANOWSKI et al. (2017). Segundo os autores, a proposta do FastText é ser uma

extensão do Skip-Gram apresentado que leva em conta a chamada informação dos fragmentos das palavras (do inglês *subword information*), com o objetivo de possibilitar a geração de boas representações para palavras que não foram utilizadas no treinamento.

No FastText, o vetor de uma palavra é representado como a soma das representações de seus fragmentos, que nada mais são do que partes dessa palavra original. Vamos analisar um exemplo para esclarecer: Se considerarmos como 3 o tamanho de cada fragmento, as partes da palavra `where` serão (após a introdução dos marcadores de início e final `<` e `>`): `<wh`, `whe`, `her`, `ere`, `re>`. Logo, o vetor que representará a palavra `where` será a combinação dos vetores de cada um desses fragmentos.

Adicionamos esses símbolos especiais que marcam o início e o final da palavra para que possamos diferenciar o que são fragmentos do que são palavras propriamente ditas. Por exemplo, a palavra `<her>` é diferente do fragmento `her`, que foi gerado durante a fragmentação de `where`.

O algoritmo por trás do FastText é essencialmente o Skip-Gram apresentado anteriormente, com algumas ligeiras modificações, em especial na hora de definir a probabilidade de uma palavra pertencente ao contexto ser gerada, dada a palavra central, como podemos ver na equação 2.5

$$p(w_c|w_t) = \frac{e^{s(w_t,w_c)}}{\sum_{w_k \in \mathbf{W}} e^{s(w_t,w_k)}} \tag{2.5}$$

onde w_c é a palavra do contexto, w_t é a palavra central e w_k são cada uma das palavras presentes em $\mathbf{W}$. Na versão original do Skip-Gram, a função $s(w_t, w_c)$ nada mais é do que o produto interno entre os vetores $\vec{w_t}$ e $\vec{w_c}$ dessas palavras. Já no FastText, nós calculamos o produto interno entre o vetor da palavra de contexto w_c e cada um dos fragmentos da palavra central w_t, da seguinte maneira:

$$s(w_t, w_c) = \sum_{f \in \mathbf{F}_t} \vec{w}_f^T \vec{w_c} \tag{2.6}$$

onde $\mathbf{F}_t$ é o conjunto de todos os fragmentos da palavra central w_t. Ou seja, a palavra t está sendo representada pela combinação de todos os seus fragmentos. Dessa maneira, como informado pelos autores, o modelo compartilha a representação dos fragmentos entre as palavras, permitindo a geração de *embeddings* de boa qualidade para palavras raras.

Como mencionado anteriormente, uma grande vantagem do FastText é a possibilidade de obtermos boas representações vetoriais para palavras que não estavam presentes durante o treinamento do algoritmo. Uma vez que essa técnica utiliza os vetores dos fragmentos de cada palavra para gerar as representações da palavra propriamente dita, é provável que os fragmentos de palavras fora do vocabulário estivessem presentes no treinamento e, dessa forma, o algoritmo tenha aprendido uma boa representação para os mesmos. Dessa maneira, combinando esses fragmentos que apareceram previamente no

treinamento, o FastText pode fornecer bons vetores para palavras não vistas no momento da geração das representações.

Essa característica é extremamente interessante, se pensarmos num cenário como as redes sociais, em que o uso de gírias, abreviações, neologismos e até mesmo palavras escritas de maneira equivocada são frequentes. Um algoritmo que seja robusto a essas variações de escrita é altamente desejável em tweets, por exemplo, que é um dos panos de fundo desse trabalho.

Definimos os espaços vetoriais, porém ainda não está estabelecida nenhuma correlação entre os dois espaços diferentes, de modo que possamos capturar as semelhanças semânticas entre as diferentes línguas. Para obtermos essa correlação, precisamos alinhar os espaços vetoriais.

3 ALINHAMENTO ENTRE ESPAÇOS VETORIAIS

Uma vez definidas as representações das palavras de cada um dos idiomas, devemos encontrar alguma maneira de gerar uma "tradução" entre os espaços vetoriais, de modo que possamos estabelecer relações sintáticas e semânticas entre palavras de idiomas diferentes. Essa maneira é justamente o alinhamento de espaços vetoriais, que abordaremos nesse capítulo.

A primeira vez que foi percebido que espaços vetoriais possuíam estruturas semelhantes entre línguas e que era possível estabelecer uma tradução entre eles foi em MIKOLOV et al. (2013b), onde os autores propuseram o aprendizado de um mapeamento linear entre um *embedding* de origem e um de destino como mecanismo de tradução.

De forma semelhante, nosso objetivo ao utilizar o alinhamento nesse trabalho é alinhar os *embeddings* num sentido geométrico, ou seja, fazer com que 2 palavras que sejam tradução uma da outra em línguas diferentes sejam representadas por pontos próximos no $\mathbb{R}^n$, após o alinhamento, trazendo as nuvens de pontos que representam as palavras de cada um dos idiomas para um mesmo lugar no espaço.

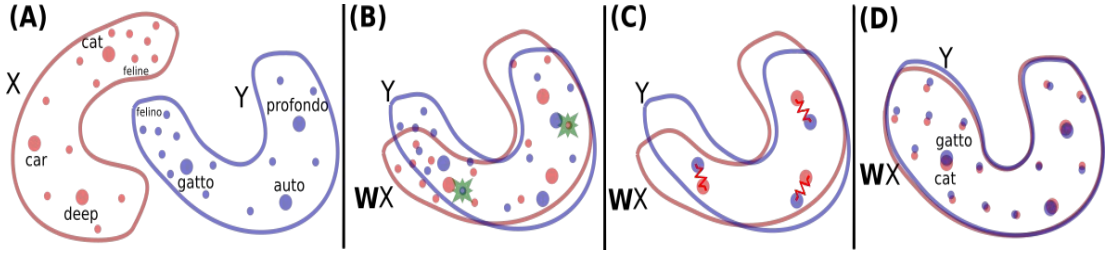
Alinhar nuvens de pontos em dimensão alta não é um problema que surgiu como meio de traduzir *word embeddings*: ideias nesse sentido já foram abordadas em visão computacional, como em COOTES et al. (1995) e TOMASI; KANADE (1992), e no processamento de linguagem natural, aplicado à própria tradução não-supervisionada (ou seja, quando não sabemos a tradução correta, *a priori*) de palavras e sentenças (RAPP, 1995). Após MIKOLOV et al. (2013b), abordagens mais recentes para essa tradução de espaços vetoriais foram propostas, com as que estão em ARTETXE; LABAKA; AGIRRE (2016) e SMITH et al. (2017).

Além de tentativas supervisionadas de alinhar espaços vetoriais, existiram estudos no sentido de diminuir e até mesmo eliminar a necessidade de conhecermos a correspondência entre as palavras, como as abordagens propostas em CAO et al. (2016) e ZHANG et al. (2017). Porém, como informado em CONNEAU et al. (2017), o desempenho dos modelos não-supervisionados era inferior ao desempenho das abordagens supervisionadas. Para contornar essa situação, nesse mesmo trabalho foi proposta uma estratégia não-supervisionada que consegue ser competitiva com os métodos que utilizam um dicionário de correspondência entre línguas, superando os resultados dessas técnicas em alguns casos.

Apresentaremos agora o método proposto em (CONNEAU et al., 2017), o MUSE, abreviação para *Multilingual Supervised and Unsupervised Embeddings*, que realizará o alinhamento entre espaços através do aprendizado de uma transformação linear W. Essa transformação será utilizada para trazer uma nuvem de pontos no espaço, que representa um idioma, para o lugar onde estão presentes as representações de outro idioma, estabelecendo o alinhamento desses pontos, como podemos observar na Figura 8.

Para aprendermos a transformação desejada, podemos empregar duas estratégias: a supervisionada, quando sabemos a correspondência entre os pontos, ou seja, quando sabemos a tradução entre as palavras dos diferentes idiomas; e a não-supervisionada, quando não conhecemos essa correspondência. Discutiremos a seguir, as principais ideias e conceitos por trás de cada uma das abordagens.

Figura 8 – Visão Geral do Alinhamento, extraída de (CONNEAU et al., 2017). **(A)** Nuvens de pontos representando os *embeddings* de diferentes idiomas. **(B)** Aplicação da transformação linear W sobre os vetores que representam o idioma X, realizando a "tradução" para o idioma Y. **(C)** Etapa de refinamento do mapeamento linear encontrado, através da geração de um dicionário sintético de correspondência entre idiomas através dos vizinhos mais próximos em cada nuvem de pontos. **(D)** Resultado final do alinhamento de espaços vetoriais, após o refinamento.



3.1 MULTILINGUAL SUPERVISED AND UNSUPERVISED EMBEDDINGS - SUPERVISIONADO

Começaremos pela maneira supervisionada de obtermos a transformação linear W que alinha os espaços vetoriais. Escrevemos nosso problema como uma minimização da diferença entre a aplicação de W sobre o idioma "fonte", que representaremos por X, que é uma matriz cujas colunas x_i denotam os vetores associados a cada uma das palavras i desse idioma, e os *embeddings* do idioma "destino", que chamaremos de Y, uma matriz da mesma forma de X. Uma vez que conhecemos a correspondência entre as palavras dos diferentes idiomas, podemos pensar no problema como essa minimização entre os erros de tradução entre os dois espaços vetoriais.

Podemos modelar o problema como:

$$W^* = \underset{W \in O_d(\mathbb{R})}{\operatorname{argmin}} \sum_{i=1}^{k} \|Wx_i - y_i\|_2^2 \tag{3.1}$$

Onde $\|\cdot\|_2$ representa a norma $L2$, d é a dimensão do espaço vetorial no qual os idiomas estão representados, k é a quantidade de palavras presentes no idioma, Wx_i é um vetor que denota a "tradução" da palavra associada ao vetor x_i para o idioma Y, e $O_d(\mathbb{R})$ é o espaço das matrizes ortogonais com números reais, que é o conjunto de matrizes que preservam normas e ângulos de vetores.

Uma vez definida a modelagem do problema, precisamos determinar alguma forma de efetivamente resolvê-lo, obtendo assim a transformação linear que desejamos. Com o emprego da restrição de ortogonalidade, nossa modelagem recai ao problema de Procrustes Ortogonal (GOWER; DIJKSTERHUIS, 2005), que apresenta uma fórmula fechada para a solução, utilizando a decomposição em valores singulares (do inglês singular value decomposition - SVD) (GOLUB; REINSCH, 1970) de YX^T.

Assim, a solução para o problema de Procrustes Ortogonal é expressa por:

$$W^* = \underset{W \in O_d(\mathbb{R})}{\operatorname{argmin}} \sum_{i=1}^{k} \|Wx_i - y_i\|_2^2 = UV^T \tag{3.2}$$

Onde UV^T é obtida através da decomposição em valores singualres $U\Sigma V^T = \text{SVD}(YX^T)$ de YX^T. Com essa solução, obteremos a transformação linear que alinha nossos espaços vetoriais. Como dito anteriormente, fomos capazes de encontrar W através dessa modelagem pois conhecíamos a correspondência entre palavras de diferentes idiomas. Analisaremos agora a abordagem não-supervisionada presente no MUSE (CONNEAU et al., 2017), que será a versão utilizada no momento de alinharmos tweets e notícias.

3.2 MULTILINGUAL SUPERVISED AND UNSUPERVISED EMBEDDINGS - NÃO-SUPERVISIONADO

Para o alinhamento dos espaços vetoriais de tweets e notícias, não dispomos de nenhum dicionário com a tradução entre as palavras pelos diferentes idiomas, inviabilizando a aplicação do método supervisionado discutido anteriormente. Logo, faz-se necessário o uso de alguma outra técnica que não necessite de nenhum tipo de tradução. Como apresentado em (CONNEAU et al., 2017), utilizaremos as chamadas redes neurais adversariais (GOODFELLOW et al., 2014) para obtermos a transformação que estabelece a tradução entre as palavras do idioma "fonte" X e as palavras do idioma "destino" Y.

As redes adversariais são uma arquitetura especial de redes neurais, sendo composta na verdade por duas redes com papeis bem definidos: a rede Geradora (do inglês *Generator*), e a Discriminadora (do inglês *Discriminator*). A rede Geradora tenta criar exemplos artificiais, que se pareçam ao máximo com os exemplos presentes no conjunto de dados original, enquanto a rede Discriminadora tem o papel de distinguir se o exemplo ao qual ela está sendo apresentada nesse momento é de fato um exemplo do conjunto de treinamento "real", ou se esse exemplo foi criado artificialmente pela rede geradora.

No nosso contexto de tradução entre idiomas, o objetivo é fazer com o que discriminador consiga distinguir se uma determinada representação vetorial veio da tradução $WX = \{Wx_1, Wx_2, \ldots, Wx_k\}$ ou do idioma Y, enquanto o gerador deve aprender a transformação linear aplicada à X, de modo que WX e Y sejam os mais similares possíveis,

fazendo com que o discriminador não consiga distinguir entre uma palavra original do idioma Y e sua "tradução" WX.

Sejam θ e W os parâmetros da rede Discriminadora e da Geradora, respectivamente. Seja K a quantidade de palavras em cada um dos idiomas. Seja ainda $D_\theta(z)$ a probabilidade do Discriminador considerar o vetor z como sendo original do idioma Y. A função objetivo da rede Discriminadora será modelada como:

$$\underset{\theta}{\operatorname{argmax}} \frac{1}{k} \sum_{i=1}^{k} \Big[\log D_\theta\Big(y_i\Big) + \log(1 - D_\theta\Big(Wx_i\Big)\Big)\Big] \tag{3.3}$$

Quando o discriminador estiver conseguindo distinguir entre os exemplos reais e os artificiais, a primeira parcela atingirá seu valor máximo, enquanto a segunda parcela irá zerar, uma vez que o gerador não estará conseguindo "enganar" o discriminador, já que ele atribuirá um valor próximo de zero para os exemplos oriundos de WX.

Já a função objetivo do Gerador será definida como:

$$\underset{W}{\operatorname{argmin}} \frac{1}{k} \sum_{i=1}^{k} \log(1 - D_\theta\Big(Wx_i\Big)\Big) \tag{3.4}$$

De maneira semelhante, quando o gerador estiver gerando *embeddings* similares às representações vetoriais originais, o discriminador não conseguirá distinguir a origem dos exemplos, e fará com que o argumento do log fique com um valor bem próximo de zero, já que o discriminador atribuirá o valor `1` a um exemplo que veio de WX.

Utilizando o método do gradiente descendente, conforme o padrão apresentado em GOODFELLOW et al. (2014), conseguiremos obter a transformação linear que efetivamente realiza a tradução entre os dois espaços vetoriais, esperando que com essa transformação consigamos estabelecer relações semânticas e sintáticas entre as diferentes línguas.

Consideraremos agora um passo adicional para o aprendizado dessa transformação de maneira não-supervisionada: como informado pelos autores do MUSE (CONNEAU et al., 2017), o resultado do alinhamento para palavras raras pode ser melhorado se realizarmos um processo de refinamento em W. Para isso, geramos um dicionário sintético de "traduções" entre palavras dos diferentes idiomas que são mutualmente vizinhos mais próximos, como ilustrado na parte **C** da Figura 8 e, em seguida, aplicamos o mesmo procedimento realizado na abordagem supervisionada, resolvendo o problema apresentado na seção 3.1.

Outra observação é que a ortogonalidade da nossa transformação linear continua sendo importante no cenário não-supervisionado. Uma matriz Q é considerada ortogonal quando suas colunas são compostas por vetores ortonormais, ou seja:

$$Q^T Q = Q Q^T = I \tag{3.5}$$

onde I é a matriz identidade. Como apresentado em (SMITH et al., 2017), é vantajoso impormos a restrição de ortogonalidade sobre nossa transformação linear, uma vez que esse tipo de transformação preserva normas de vetores, assim como distâncias. Dessa forma, ao alinharmos os dois espaços vetoriais, teremos a garantia que as relações previamente existentes entre as palavras de um mesmo idioma serão preservadas.

Levando essa vantagem em consideração, os autores propuseram uma regra adicional para a otimização da transformação linear, com o objetivo de preservar a ortogonalidade da mesma, inspirada em CISSE et al. (2017), onde foi proposto que, após cada atualização do parâmetro W do Gerador pelo gradiente descendente, fosse garantida a ortogonalidade do mesmo, através da inserção da seguinte regularização à W, onde β é um hiper-parâmetro:

$$R(W) = \frac{\beta}{2}\|W^T W - I\|_2^2 \tag{3.6}$$

Utilizamos o gradiente descendente para resolver essa etapa adicional da atualização de W. Em (CONNEAU et al., 2017), os autores recomendam um valor para β em torno de 0.01. Em nossos experimentos, nós testamos alguns valores diferentes para esse hiper-parâmetro.

Uma vez que alinhamos os espaços de representação das palavras e portanto estabelecemos a correlação entre os diferentes idiomas, precisamos de alguma ferramenta para efetivamente determinar a similaridade entre dois documentos, ou seja, geometricamente necessitamos de alguma forma de mensurar a distância entre esses documentos.

4 DISTÂNCIAS ENTRE DOCUMENTOS

Para determinarmos similaridade entre documentos, sejam eles páginas da Wikipedia, artigos científicos, tweets ou notícias, precisamos definir alguma maneira de estabelecer uma distância, ou seja, um grau de similaridade entre diferentes documentos. Mas antes de introduzirmos a distância propriamente dita, precisamos estabelecer a maneira pela qual representaremos os nossos documentos, os tweets e notícias.

4.1 REPRESENTAÇÃO DOS DOCUMENTOS

Um dos primeiros métodos propostos para a representação de documentos foi o Modelo de Espaço Vetorial (do inglês *Vector Space Model*) (SALTON; WONG; YANG, 1975), que consistia em modelar um documento utilizando a frequência dos termos, oriundos de um vocabulário pré-definido, presentes no mesmo. Duas representações existentes para esse modelo são o TF e o IDF.

A primeira representação, o TF, consiste numa contagem da frequência das palavras num determinado documento. Dessa maneira, nosso texto será representado por um vetor do tamanho do vocabulário pré-fixado, onde cada posição do vetor, que corresponde a um determinado termo, conterá o número de vezes em que a mesma apareceu naquele documento.

A segunda representação, o IDF, pode ser encarada como uma extensão do TF, onde levamos em conta a quantidade de vezes em que um determinado termo apareceu por todos os nossos documentos. O objetivo dessa representação é penalizar palavras que aparecem em vários documentos, uma vez que palavras muito frequentes entre textos não agregarão tanto significado para o documento em si, ao passo que palavras que sejam bem específicas para um determinado texto carreguem um maior peso semântico.

Para essa representação, cada posição do vetor que representa um documento receberá o seguinte valor:

$$d_i = c_i * log\left(\frac{N+1}{df(i)+1}\right) \tag{4.1}$$

onde c_i é o número de vezes em que a palavra i apareceu no documento d; N é a quantidade de documentos na nossa base de dados e $df(i)$ é a quantidade de documentos em que a palavra i está presente.

Utilizando esse tipo de representação esparsa, é possível calcular distância entre documentos, através de técnicas como a similaridade cosseno por exemplo. Porém esse tipo de representação não codifica informação semântica em suas dimensões, sendo inadequado para estabelecermos similaridade entre documentos, como observado em KUSNER et al.

(2015), onde os autores propuseram o seguinte exemplo pra ilustrar essa observação: queremos determinar a similaridade semântica entre as seguintes frases: "Obama speaks to the media in Illinois" e "The President greets the press in Chicago". Após realizarmos a remoção das chamadas *stop words*, que são as palavras que não carregam muito significado por si só, como artigos e preposições, ficaremos com os documentos "Obama speaks media Illinois" e "President greets press Chicago".

Digamos que a nossa base de dados seja formado apenas por essas duas frases. O vocabulário que as mesmas geram será formado por `[Obama, President, speaks, greets, media, press, Illinois, Chicago]`. Se representarmos os documentos d_1 e d_2 utilizando a representação TF, apresentada anteriormente, obteremos os vetores `[1, 0, 1, 0, 1, 0, 1, 0]` e `[0, 1, 0, 1, 0, 1, 0, 1]`. Uma similaridade cosseno calculada sobre essa representação nos indicará que os documentos são completamente "ortogonais", ou seja, esses documentos não possuem semelhança nenhuma; o que não é verdade, se estivermos analisando apenas as palavras que as compõem.

Portanto, é necessário o emprego de outro tipo de representação de documentos que consiga codificar informações de significado em suas componentes. Assim, utilizaremos os espaços vetoriais de palavras, apresentado e discutido nos capítulos anteriores, como a nova forma de representar os documentos que consegue agregar informação semântica.

Para esse caso, os documentos serão modelados como uma nuvem de pontos ponderada no espaço dos *word embeddings*, onde cada ponto representa um termo presente naquele documento e os pesos em cada ponto são expressos pela representação daquele documento num dos Modelos de Espaço Vetorial apresentados anteriormente.

Para calcularmos a similaridade entre documentos representados nesse espaço dos *embeddings*, iremos recorrer à distância que apresentaremos a seguir, a *Word Mover's Distance*, introduzida por KUSNER et al. (2015).

4.2 WORD MOVER'S DISTANCE

A *word mover's distance* (WMD), um caso especial *Earth Mover's Distance* (EMD) (RUBNER; TOMASI; GUIBAS, 1998), é uma distância calculada sobre o espaço vetorial de palavras, que será usada para determinar o quão custoso é nos transportarmos de uma determinada nuvem de pontos (que representa um documento) até outra.

Seja Φ a matriz de fluxo entre os documentos P e Q, onde o elemento Φ_{ij} nos indica o quanto da palavra presente na posição i da distribuição P, deve ser transportada para a palavra no posição j em Q . Seja C a matriz de distâncias entre uma palavra na posição i de P, denotada por p_i, e outra na posição j de Q, denotada por q_j. Seja $C \circ \Phi$ o produto de Hadamard, que consiste na multiplicação elemento por elemento entre essas matrizes.

Vamos considerar ainda três restrições que essa distância apresenta: a primeira delas é que para transformarmos totalmente o documento P em Q, a quantidade do fluxo

que sai de um ponto arbitrário em P e chega na posição j de Q deve satisfazer o valor q_j. A segunda restrição nos impõe que o fluxo que saiu de um determinado ponto p_i e chegou a um ponto arbitrário em Q deve ser igual ao valor original p_i. Finalmente, a última restrição indica que todo elemento de Φ deve ser não-negativo, indicando que o transporte deve ocorrer sempre do documento P para o documento Q, e nunca no sentido inverso.

Dessa forma, a WMD é definida como:

$$\begin{aligned} \min_{\Phi} \|C \circ \Phi\|_2^2 \text{ t. q.} \\ \sum_i \Phi_{ij} = q_j \\ \sum_j \Phi_{ij} = p_i \\ \Phi_{ij} \geq 0 \end{aligned} \tag{4.2}$$

O custo para movermos uma palavra para outra será justamente a distância euclidiana entre os pontos no espaço vetorial que representam cada uma das palavras. Seja w_i o vetor associado à i-ésima palavra do documento P e w_j o vetor associado à j-ésima palavra do documento Q. Dessa forma, o custo c_{ij} de sairmos da palavra i para a palavra j, será dada por $\|\vec{w_i} - \vec{w_j}\|_2$.

Ou seja, quanto mais semanticamente parecidas foram as duas palavras, menor será a distância entre elas, e portanto menos custoso será o transporte entre termos que apresentam esse tipo de semelhança. Como alinhamos previamente os espaços vetoriais, para realizarmos a "tradução" entre as palavras dos diferentes idiomas, esperamos que as similaridades entre línguas estejam estabelecidas, e que a matriz de custos consiga representar bem a similaridade entre dois idiomas.

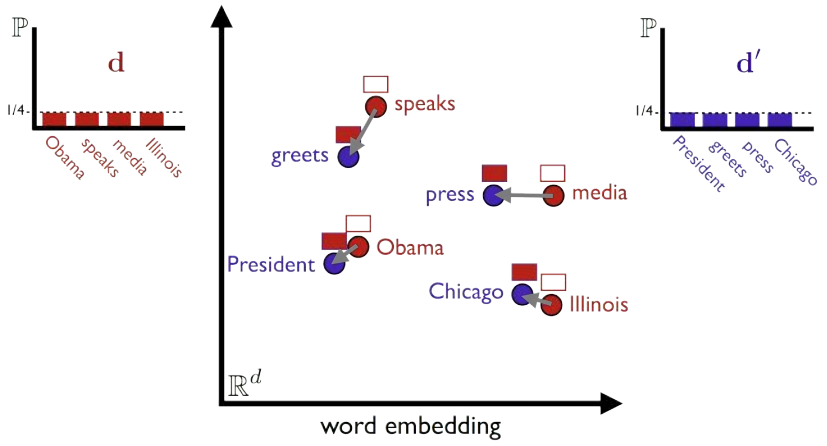
Naturalmente, o custo de movermos uma palavra para outra será a unidade básica para estabelecer a distância entre dois documentos, buscando minimizar o custo total de transportar todas as palavras. Retomando o exemplo apresentado na seção 4.1, podemos visualizar o transporte entre as palavras de cada documento na Figura 9.

Como podermos perceber, é bem mais barato movermos a palavra "speaks" para "greets" do que para qualquer uma das outras palavras, já que essas outras possuem uma similaridade semântica bem menor. De maneira semelhante, é mais barato movermos "Chicago" para "Illinois" do que para "speaks".

Realizando esse transporte entre palavras similares do ponto de visto semântico, é esperado que a WMD nos informe que a distância entre esses documentos seja pequena, ficando comprovada a semelhança semântica entre os dois textos.

Finalmente, para obtermos o plano de transporte ótimo, devemos resolver o problema de programação linear que define a WMD. Para isso, pode-se utilizar o método simplex ou

Figura 9 – Representação de documentos como nuvem de pontos no espaço vetorial dos Word Embeddings. As setas representam a estratégia ótima de transporte dos pontos de um documento ao outro. Extraída de (NICULAE, 2015)



o método de pontos interiores, ou ainda métodos especializados em resolver o problema de calcular a distância de Wasserstein entre duas distribuições, como o apresentado em (PELE; WERMAN, 2009).

4.3 DIVERGÊNCIA ENTRE DISTRIBUIÇÕES

Se interpretarmos os documentos como distribuições de probabilidade no espaço dos *embeddings*, a WMD coincide com a distância de Wasserstein (VILLANI, 2008), utilizada para determinar o grau de similiaridade entre distribuições. Existem outras formas de calcular esse tipo de informação, como a divergência de Kullback-Leibler (KULLBACK; LEIBLER, 1951) e a divergência de Jensen-Shannon (FUGLEDE; TOPSOE, 2004). Vamos definí-las a seguir, e realizar um comparativo entre elas, com o intuito de determinarmos se existe alguma vantagem em utilizarmos uma em detrimento das outras.

4.3.1 Divergência de Kullback-Leibler

A divergência de Kullback-Leibler, ou divergência KL (KULLBACK; LEIBLER, 1951), é uma técnica que nos permite medir o quanto uma determinada distribuição p diverge de outra distribuição q. Podemos calcular essa divergência da seguinte maneira:

$$D_{KL}(p \parallel q) = \int_x p(x) \log \frac{p(x)}{q(x)} dx \tag{4.3}$$

onde x representa cada um dos pontos para os quais a probabilidade está definida. Essa divergência atinge seu mínimo quando $p(x) = q(x)$ para todo x. Outro fato que vale a pena ser comentado é que, se para um determinado x, $q(x) = 0$ e $p(x) > 0$, essa divergência vai para $+\infty$.

4.3.2 Divergência de Jensen-Shannon

Uma outra medida de similaridade entre distribuições é a chamada Divergência de Jensen-Shannon (FUGLEDE; TOPSOE, 2004). Ao contrário da divergência KL, essa medida possui valores limitados ao intervalo $[0, 1]$, é simétrica e apresenta uma maior suavidade. Podemos definí-la como:

$$D_{JS}(p \parallel q) = \frac{1}{2} D_{KL}\left(p \,\middle\|\, \frac{p+q}{2}\right) + \frac{1}{2} D_{KL}\left(q \,\middle\|\, \frac{p+q}{2}\right) \tag{4.4}$$

4.4 COMPARATIVO ENTRE DIVERGÊNCIAS E DISTÂNCIA DE WASSERSTEIN

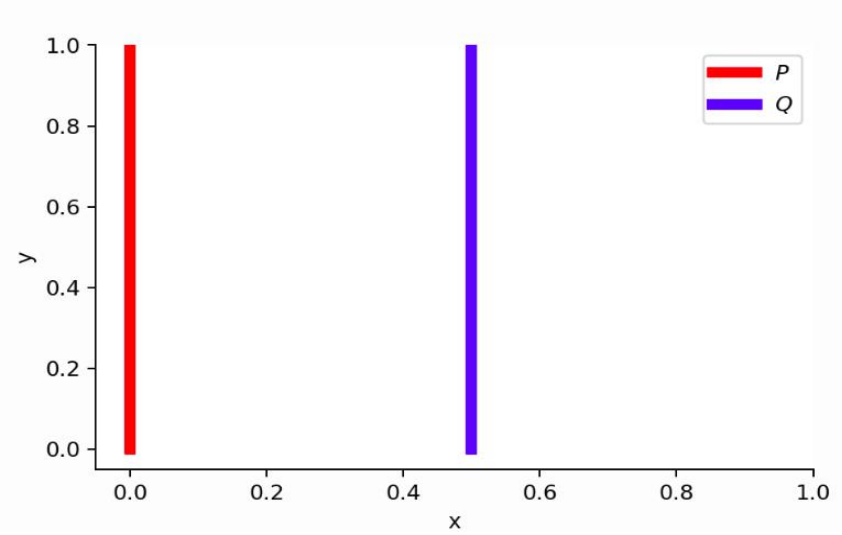
Vamos basear nossa, discussão no seguinte exemplo, apresentado em ARJOVSKY; CHINTALA; BOTTOU (2017): consideremos duas distribuições P e Q no $\mathbb{R}^2$, ilustradas na Figura 10 e definidas como:

$$\forall (x, y) \in P, x = 0 \text{ e } y \sim U(0, 1) \tag{4.5}$$

$$\forall (x, y) \in Q, x = \theta \text{ e } y \sim U(0, 1) \tag{4.6}$$

onde $U(a, b)$ representa a distribuição uniforme no intervalo $[a, b]$.

Figura 10 – Distribuições P e Q, com $\theta = 0.5$. Para $\theta \neq 0$, não há sobreposição entre as distribuições. Extraído de (WENG, 2017)



Utilizaremos cada uma das técnicas apresentadas para estimarmos a distância entre as duas probabilidades.

- KL: Como comentado anteriormente, se existe um ponto (x, y) tal que $P(x, y) > 0$, mas $Q(x, y) = 0$, a divergência KL irá para $+\infty$. Logo,

$$D_{KL}(P \parallel Q) = \begin{cases} +\infty & \text{se } \theta \neq 0 \\ 0 & \text{se } \theta = 0 \end{cases} \tag{4.7}$$

- JS: Para todo ponto (x, y) tal que $P(x, y) \neq 0$, o "segundo termo" da divergência de Jensen-Shannon será $M = \frac{1}{2}P$. Vamos analisar a primeira parcela da equação:

$$\begin{aligned} D_{KL}(P \parallel M) &= \int_{(x,y)} P(x,y) \log\left(\frac{P(x,y)}{M(x,y)}\right) dx\, dy \\ &= \int_{(x,y)} P(x,y) \log\left(\frac{P(x,y)}{\frac{P(x,y)}{2}}\right) dx\, dy \\ &= \int_{(x,y)} P(x,y) \log(2)\, dx\, dy \\ &= \log(2) \end{aligned} \tag{4.8}$$

  De maneira análoga, $D_{KL}(Q \parallel M) = \log(2)$. Assim,

$$D_{JS}(P \parallel Q) = \begin{cases} \log(2) & \text{se } \theta \neq 0 \\ 0 & \text{se } \theta = 0 \end{cases} \tag{4.9}$$

- Wasserstein: Como a diferença entre as duas distribuições é apenas uma translação, a melhor maneira de realizarmos o transporte de P para Q é descolarmos a massa numa linha reta de $(0, y)$ para (θ, y). Portanto, $W(P, Q) = |\theta|$.

Intuitivamente, podemos pensar que, conforme $\theta \to 0$, a distância entre P e Q deve diminuir proporcionalmente. Como podemos perceber, para as divergências de Kullback-Leibner e Jensen-Shannon, não é isso o que acontece. Apenas a distância de Wasserstein conseguiu nos fornecer uma informação precisa e interpretável, no que diz respeito à distância entre as distribuições.

Com esse exemplo simples, podemos evidenciar a qualidade e interpretabilidade do grau de similaridade entre duas distribuições obtido pela distância de Wasserstein. Mas claro, as vantagens dessa distância em relação às outras divergências também se aplicam a distribuições em dimensão mais alta e com maior interseção entre si. Como a demonstração dessas vantagens foge ao escopo desse trabalho, citaremos apenas a referência (ARJOVSKY; CHINTALA; BOTTOU, 2017) e discutiremos seus resultados mais importantes.

O primeiro deles diz respeito às garantias de continuidade e diferenciabilidade da distância de Wasserstein, que são propriedades muito interessantes e extremamente importantes quando trabalhamos com algoritmos de otimização. As divergências KL e JS não possuem essas garantias, o que pode prejudicar a interpretabilidade e otimização de problemas que as utilizem como funções de custo.

O segundo prova que qualquer par de distribuições, para o qual é possível convergir de uma distribuição para outra, com relação às divergências KL e JS também converge pela

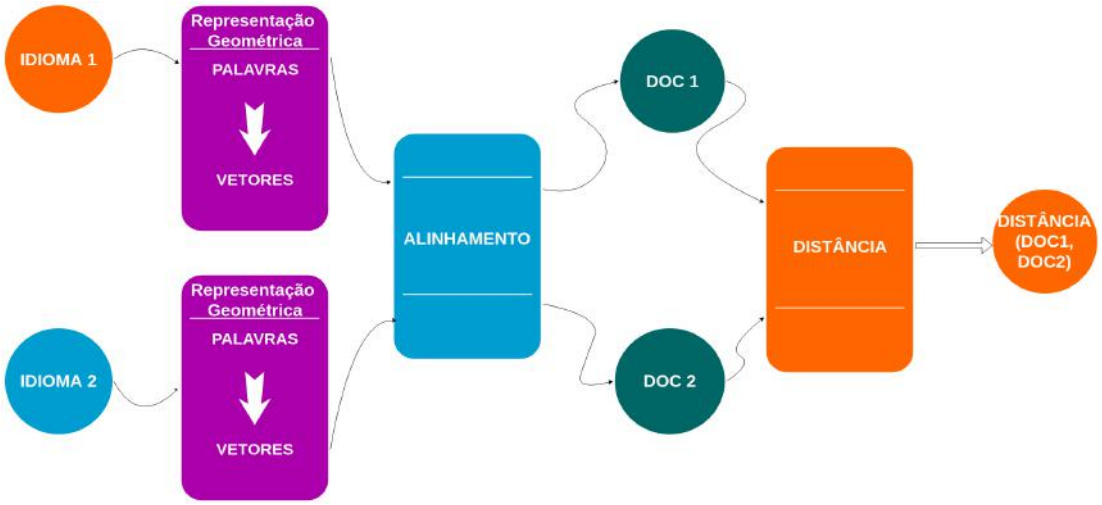
distância de Wasserstein. Além disso, foi provado que uma pequena distância de Wasserstein de fato corresponde a uma pequena diferença entre as distribuições, evidenciando mais uma vez a interpretabilidade dessa distância.

Uma vez estabelecida a última etapa da abordagem, poderemos aplicá-lo ao conjunto de dados escolhido, evidenciando a metodologia aplicada e os resultados obtidos.

5 METODOLOGIA

Nos capítulos anteriores foram apresentadas e discutidas cada uma das etapas da abordagem, ilustrada na Figura 11, enfatizando a importância de cada elas para o funcionamento da mesma como um todo. Discutiremos agora a metodologia adotada nesse trabalho.

Figura 11 – Blocos Constituintes da formulação da abordagem



Iniciaremos os experimentos verificando a reprodutibilidade dos experimentos originais, que estão presentes (BALIKAS et al., 2018), e em seguida iniciaremos uma série de modificações incrementais em cada um dos componentes da nossa formulação, com a finalidade de garantir a robustez da abordagem apresentada e comprovar a adaptabilidade do método para tarefas diferentes.

Para a primeira etapa iremos utilizar os mesmos espaços vetoriais e base de dados empregadas, com o objetivo de comprovar o desempenho do método na tarefa de recuperar a versão correspondente em francês de uma determinada página da Wikipedia em inglês, e vice-versa (recuperar a versão em inglês para uma página da Wikipedia em francês).

A segunda etapa será a modificação dos espaços vetoriais utilizados nos experimentos originais, que é onde se iniciam as nossas mudanças incrementais no método. No artigo base, os *word embeddings* utilizados para o inglês e para o francês são do Numberbatch (SPEER; LOWRY-DUDA, 2017), já em nossos experimentos foram utilizados *embeddings* pré-treinados do FastText (BOJANOWSKI et al., 2017) para cada um dos respectivos idiomas.

Nosso objetivo nessa etapa é verificar o desempenho do método quando modificamos a primeira etapa da abordagem, que é justamente o bloco que estabelece as relações sintáticas e semânticas entre as palavras de um mesmo idioma.

Porém não basta simplesmente alterarmos o método para utilizar esses novos espaços vetoriais, pois ainda não temos estabelecido um componente essencial para a abordagem: as relações de significado entre palavras de diferentes idiomas. Com isso, deveremos obter essas relações, empregando o método de alinhamento de espaços vetoriais discutido nos capítulos anteriores, o MUSE (CONNEAU et al., 2017), utilizando as duas versões disponíveis (supervisionado e não-supervisionado).

Os *embeddings* do NumberBatch utilizados no artigo base (BALIKAS et al., 2018) não necessitam de alinhamento uma vez que, segundos os próprios autores (SPEER; LOWRY-DUDA, 2017), as representações de diferentes idiomas estão dispostas sobre uma localização comum no espaço vetorial que representa as palavras.

A terceira e última etapa será a modificação do conjunto de dados sobre o qual a abordagem é aplicada e finalmente resolver a tarefa que foi proposta para esse trabalho. Agora, não tentaremos mais recuperar versões em idiomas diferentes de páginas da Wikipedia, e sim encontrar os tweets mais relacionados a cada uma das notícias da nossa base de dados, verificando assim o desempenho do nosso método para essa tarefa, e comparando os resultados obtidos com aqueles reportados no artigo original (SUAREZ et al., 2018)

Para essa etapa, utilizaremos duas configurações de *word embeddings*: na primeira utilizaremos os espaços vetoriais treinados sobre as bases de dados de notícias e tweets separadamente, que estão disponíveis em SUAREZ et al. (2018). A motivação para nossa decisão de testarmos *embeddings* separados para tweets e notícias é explorar a hipótese que essas fontes de informação podem ser interpretadas como "idiomas" diferentes, devido à pouca semelhança de vocabulário, conforme apresentado em (CHEN, 1994) e (FURNAS et al., 1987).

A segunda será utilizar o mesmo espaço vetorial para tweets e notícias, a saber, o espaço vetorial em inglês pré-treinado pelo FastText (BOJANOWSKI et al., 2017).

Com essa abordagem, poderemos comprovar a hipótese de que faz sentido gerar espaços vetoriais para esses dois universos de informação diferentes, uma vez comprovada a superioridade de desempenho em relação ao espaço vetorial treinado sem um foco específico na base de dados, como foi apresentado em LI et al. (2017a).

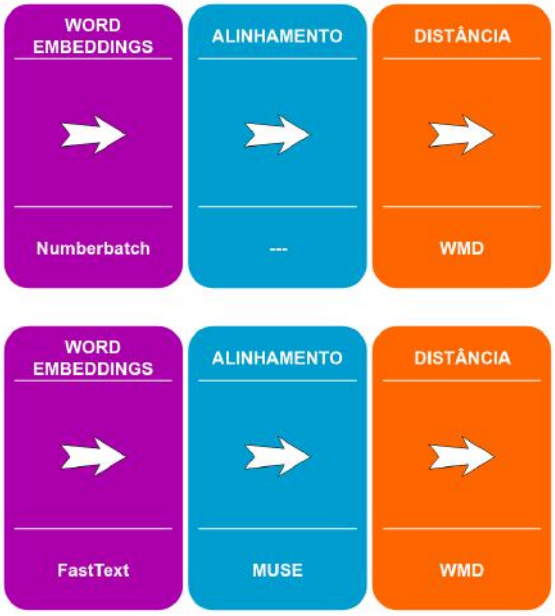
Assim como realizado na etapa anterior, deveremos realizar o alinhamento dos *word embeddings* de tweets e notícias, utilizando o MUSE (CONNEAU et al., 2017). Agora, utilizaremos apenas a versão não-supervisionada, visto que não possuímos a correspondência entre palavras desses 2 "idiomas" distintos. Em TSAGKIAS; RIJKE; WEERKAMP (2011), os autores realizaram uma análise entre o vocabulário das redes sociais e dos portais de notícias utilizando a divergência de Kullback-Leibler (KULLBACK; LEIBLER, 1951) e puderam comprovar a grande diferença entre as palavras desses 2 universos de informação.

Nós utilizamos algumas configurações de hiper-parâmetros diferentes para o alinhamento, e testamos algumas recomendações presentes em CONNEAU et al. (2017), como

a centralização dos vetores do espaço, que consiste em subtrair dos vetores a média de cada uma das dimensões. Não foi explicitada o motivo para essa recomendação, porém decidimos testá-la mesmo assim, para comparar o seu desempenho com os outros espaços alinhados sem nenhuma modificação.

A Figura 12, onde o diagrama da nossa formulação foi ligeiramente comprimido (para melhor visualização das diferenças entre os blocos), nos resume as principais diferenças entre as duas abordagens que, mais precisamente, estão nos primeiros 2 blocos, onde modificamos a representação vetorial de palavras utilizada e a técnica de alinhamento empregada.

Figura 12 – Principais diferenças entre a formulação original e a utilizada nesse trabalho



Antes de iniciarmos a apresentação e discussão dos resultados, existe um outro ponto a ser citado: No experimento original, a acurácia estava sendo utilizada como medida de performance; na nossa abordagem, utilizaremos a precisão em k (P@k), que foi a medida utilizada no artigo da base de dados (SUAREZ et al., 2018).

Na abordagem original, a preocupação é que o método retorne a página (em outro idioma) correspondente a página no idioma original. Nesse caso, existe uma correspondência única entre as páginas nos diferentes idiomas, e desejamos que o algoritmo retorne exatamente essa página.

Já no problema de relacionar tweets e notícias, existe um número arbitrário de tweets que podem estar relacionados e ser relevantes para uma determinada notícia. Portanto, é necessário que a nossa medida de performance possa considerar uma correspondência maior entre a entrada e as saídas esperadas. Assim, utilizaremos a P@K, que pode ser

definida como segue na equação 5.1:

$$\text{P@k} = \frac{\#\text{ itens relevantes dentre recomendados}}{\#\text{ itens recomendados}} \tag{5.1}$$

onde k é a quantidade de itens recomendados. Realizando essa modificação, poderemos avaliar adequadamente o desempenho do método proposto, e compará-lo com os resultados originais.

Uma vez que a metodologia foi apresenta, apresentaremos e discutiremos os resultados obtidos para cada uma das etapas da série de modificações.

6 RESULTADOS

Iniciaremos agora a análise dos resultados obtidos para cada uma das etapas da série de experimentos. Como mencionado no capítulo anterior, a primeira etapa dos testes realizados foi verificar a reprodutibilidade dos resultados apresentados no artigo original (BALIKAS et al., 2018), utilizando os mesmos dados. Os resultados dessa etapa estão presentes nas tabelas 2 e 3.

A primeira coluna representa o esquema de pesos utilizados na modelagem de documentos no espaço dos *word embeddings*. Utilizamos a notação $l_1 \rightarrow l_2$ para indicar a direção em que a recuperação de páginas da Wikipedia foi feita. No caso geral, essa notação significa que estamos buscando documentos no idioma l_1 dados documentos no idioma l_2 como entrada para o método. Na Tabela 2, constam os resultados para a recuperação de páginas em inglês, dados documentos em Francês, e na Tabela 3 temos os resultados na direção oposta.

Tabela 2 – Reprodução dos experimentos originais - En → Fr

Esquema de Pesos	Reportados	Experimentos	Diferença Absoluta
TF	**0,744**	0,742	0,002
IDF	**0,778**	0,778	0,000

Medida utilizada: Acurácia

Tabela 3 – Reprodução dos experimentos originais - Fr → En

Esquema de Pesos	Reportados	Experimentos	Diferença Absoluta
TF	**0,748**	0,747	0,001
IDF	0,784	**0,788**	0,004

Medida utilizada: Acurácia

Como podemos perceber, a diferença entre os resultados reportados e os obtidos em nossos experimentos foi mínima, comprovando que o artigo original é reproduzível com os dados disponibilizados pelos autores.

Na segunda etapa de testes, o objetivo é verificar como o método se comporta quando modificamos a parte responsável pelo estabelecimento de relações semânticas e sintáticas, a saber, *os word embeddings*. Intuitivamente, podemos imaginar que um espaço vetorial bem treinado conseguirá capturar essas relações, seja qual for o algoritmo que tenha sido utilizado para treiná-lo. Logo, espera-se que os resultados obtidos utilizando os espaços vetoriais treinados com o FastText (BOJANOWSKI et al., 2017) e com o Numberbatch (SPEER; LOWRY-DUDA, 2017) apresentem desempenhos semelhantes.

Como previamente comentado na seção anterior, precisamos realizar o alinhamento dos *word embeddings* de diferentes idiomas, para estabelecermos as relações de semântica e sintaxe entre as duas línguas. Utilizando o MUSE (CONNEAU et al., 2017) nas versões supervisionada e não-supervisionada, nós realizamos o alinhamento considerando as duas direções possíveis entre fonte e destino. A notação $l_1 \rightarrow l_2$ continuará a ser utilizada nessa etapa e, além dela, introduziremos uma nova: $l_1 \rightsquigarrow l2$ representa a direção em que o alinhamento foi realizado (por exemplo, En $\rightsquigarrow$ Fr indica que os *word embeddings* do Inglês foram utilizados como fonte, e os do Francês foram o destino)

Nas tabelas 4 e 5, constam os resultados da aplicação do alinhamento supervisionado, utilizando o espaço vetorial em inglês como fonte e, naturalmente, o espaço vetorial em francês como destino. Na Tabela 4 estão os resultados para a recuperação de documentos em inglês, e na Tabela 5 para a recuperação em francês.

Tabela 4 – Modificação dos espaços vetoriais - Alinhamento Supervisionado: En $\rightsquigarrow$ Fr - En $\rightarrow$ Fr

Esquema de Pesos	Reportados	Experimentos	Diferença Absoluta
TF	0,744	**0,755**	0,011
IDF	0,778	**0,788**	0,010

Medida utilizada: Acurácia

Nas tabelas 6 e 7, constam os resultados da aplicação do alinhamento supervisionado, utilizando o francês como fonte e o inglês como destino. Na Tabela 6 estão os resultados para a recuperação de documentos em inglês, e na Tabela 7 para a recuperação em francês.

A partir desse ponto, estaremos utilizando o alinhamento não supervisionado nos experimentos dessa etapa. As tabelas 8 e 9 indicam os resultados para a recuperação de documentos em inglês e em francês, respectivamente, quando realizamos o alinhamento do inglês para o francês.

Nas tabelas 10 e 11 constam os resultados para o alinhamento utilizando o francês como fonte e o inglês como destino.

Tabela 5 – Modificação dos espaços vetoriais - Alinhamento Supervisionado: En $\rightsquigarrow$ Fr - Fr $\rightarrow$ En

Esquema de Pesos	Reportados	Experimentos	Diferença Absoluta
TF	**0,748**	0,744	0,004
IDF	**0,784**	0,774	0,010

Medida utilizada: Acurácia

Tabela 6 – Modificação dos espaços vetoriais - Alinhamento Supervisionado: Fr $\rightsquigarrow$ En - En $\rightarrow$ Fr

Esquema de Pesos	Reportados	Experimentos	Diferença Absoluta
TF	0,744	**0,748**	0,004
IDF	**0,778**	0,775	0,003

Medida utilizada: Acurácia

Tabela 7 – Modificação dos espaços vetoriais - Alinhamento Supervisionado: Fr $\rightsquigarrow$ En - Fr $\rightarrow$ En

Esquema de Pesos	Reportados	Experimentos	Diferença Absoluta
TF	**0,748**	0,739	0,009
IDF	**0,784**	0,768	0,016

Medida utilizada: Acurácia

Tabela 8 – Modificação dos espaços vetoriais - Alinhamento Não-Supervisionado: En $\rightsquigarrow$ Fr - En $\rightarrow$ Fr

Esquema de Pesos	Reportados	Experimentos	Diferença Absoluta
TF	0,744	**0,753**	0,009
IDF	**0,778**	0,778	0,000

Medida utilizada: Acurácia

Tabela 9 – Modificação dos espaços vetoriais - Alinhamento Não-Supervisionado: En $\rightsquigarrow$ Fr - Fr $\rightarrow$ En

Esquema de Pesos	Reportados	Experimentos	Diferença Absoluta
TF	**0,748**	0,743	0,005
IDF	**0,784**	0,775	0,009

Medida utilizada: Acurácia

Tabela 10 – Modificação dos espaços vetoriais - Alinhamento Não-Supervisionado: Fr $\rightsquigarrow$ En - En $\rightarrow$ Fr

Esquema de Pesos	Reportados	Experimentos	Diferença Absoluta
TF	0,744	**0,751**	0,007
IDF	0,778	**0,781**	0,003

Medida utilizada: Acurácia

Tabela 11 – Modificação dos espaços vetoriais - Alinhamento Não-Supervisionado: Fr $\rightsquigarrow$ En - Fr $\rightarrow$ En

Esquema de Pesos	Reportados	Experimentos	Diferença Absoluta
TF	**0,748**	0,744	0,004
IDF	**0,784**	0,777	0,007

Medida utilizada: Acurácia

Como podemos observar em cada uma dessas tabelas correspondentes à segunda etapa da série de experimentos, o método apresentou aproximadamente o mesmo desempenho em todos os testes, sendo que em alguns deles os resultados reportados no artigo original (BALIKAS et al., 2018) foram ligeiramente superados. Todos esses resultados nos mostram que a qualidade dos *word embeddings* é uma peça fundamental para a eficiência do método, uma vez que se alterarmos esse bloco da abordagem, substituindo-o por um espaço vetorial bem treinado, o desempenho será semelhante. Além disso, fica comprovada a eficiência do método de alinhamento empregado, no que diz respeito a capturar as semelhanças sintáticas e semânticas entre idiomas diferentes, não importando em qual direção estejamos realizando-o.

Iniciando a terceira e última etapa de nossos experimentos, apresentaremos os resultados obtidos considerando a mudança no conjunto de dados sobre o qual aplicamos o método. Como mencionado anteriormente, agora queremos recuperar os tweets mais relevantes dada uma determinada notícia. Utilizaremos os espaços vetoriais que foram gerados utilizando as bases de tweets e notícias presentes em (SUAREZ DYAA ALBAKOUR; ESQUIVEL, 2018). Vale ressaltar que os dados que compõem o objeto principal dessa etapa (as relações entre tweets e notícias) foram extraídos dessas mesmas bases.

Utilizamos duas escolhas de *word embeddings*: a primeira delas foi gerar os espaços vetoriais utilizando as bases de dados fornecidas pelos autores do artigo do conjunto de dados. Foram gerados dois espaços, um para as notícias e outro para os tweets. Em seguida, esses espaços foram alinhados utilizando o MUSE (CONNEAU et al., 2017) com algumas combinações de hiper-parâmetros.

Algumas dessa combinações, assim como os resultados obtidos quando utilizamos cada uma delas no alinhamento, estão presentes entre as Tabelas 12 e 15. Em cada uma dessas tabelas, estão explicitadas a direção em que o alinhamento foi realizado assim como os hiper-parâmetros que tiveram o seu valor padrão utilizado.

Tabela 12 – Modificação do conjunto de dados - Alinhamento Não-Supervisionado: Tweets $\rightsquigarrow$ Notícias

Abordagem Considerada	Precisão @ 5	Precisão @ 10
SUAREZ et al. (2018)	**0,59**	**0,55**
Nossa - TF	0,56	0,50
Nossa - IDF	0,58	0,55

Hiper-parâmetros modificados: `map_id_init=False, beta=0.01`

Tabela 13 – Modificação do conjunto de dados - Alinhamento Não-Supervisionado: Tweets ⇝ Notícias

Abordagem Considerada	Precisão @ 5	Precisão @ 10
SUAREZ et al. (2018)	0,59	**0,55**
Nossa - TF	0,56	0,51
Nossa - IDF	**0,61**	0,55

Hiper-parâmetros modificados: `map_id_init=False, beta=0.01, norm_emb=‘‘center’’`

Tabela 14 – Modificação do conjunto de dados - Alinhamento Não-Supervisionado: Tweets ⇝ Notícias

Abordagem Considerada	Precisão @ 5	Precisão @ 10
SUAREZ et al. (2018)	0,59	0,55
Nossa - TF	0,56	0,51
Nossa - IDF	**0,60**	**0,56**

Hiper-parâmetros modificados: `map_id_init=False, norm_emb=‘‘center’’`

Tabela 15 – Modificação do conjunto de dados - Alinhamento Não-Supervisionado: Notícias ⇝ Tweets

Abordagem Considerada	Precisão @ 5	Precisão @ 10
SUAREZ et al. (2018)	**0,59**	**0,55**
Nossa - TF	0,55	0,50
Nossa - IDF	0,59	0,55

Como podemos perceber, o desempenho do método com relação aos resultados reportados foi ultrapassado, para determinadas configurações de hiper-parâmetros.

Vamos apresentar agora, na Tabela 16 o desempenho do método quando utilizamos a segunda escolha de *embeddings*, a saber, utilizar o mesmo espaço vetorial tanto para tweets quanto para notícias.

Tabela 16 – Modificação do conjunto de dados - Comparação de desempenho utilizando o mesmo *word embedding*

Abordagem considerada	Precisão @ 5	Precisão @ 10
SUAREZ et al. (2018)	0,59	0,55
Nossa - TF	0,73	0,67
Nossa - IDF	**0,74**	**0,69**

Como podemos notar, o desempenho apresentado foi muito superior em relação aos resultados originais, e também foi superior aos resultados apresentados quando utilizamos espaços vetoriais treinados utilizando as bases de dados de tweets e notícias disponibilizadas no artigo do conjunto de dados.

Com essas observações, podemos afirmar que a qualidade do *word embedding* influencia muito no resultado final do método de recuperação. De fato, uma vez que o desempenho do método se baseia nas relações sintáticas e semânticas capturadas pelos espaços vetoriais de palavras, quanto melhor for o *embedding*, melhor será o resultado apresentado.

Existem alguns fatores que podem influencias na qualidade dos *word embeddings*. Um dos fatores é o tamanho da base de dados utilizada para treinar os espaços. No caso do FastText, foi utilizado um *dump* de todo o conteúdo da Wikipedia em inglês (BOJANOWSKI et al., 2017), que conta com aproximadamente 6 milhões de páginas. Em nossos experimentos, nós utilizamos 1 milhão de notícias para treinar o *embedding* correspondente, o que já é uma diferença bastante significativa. Espera-se que quanto maior for o número de documentos utilizados no treinamento, maiores serão as ocorrências das diferentes relações de significado e sintaxe presentes nos idiomas, e, portanto, maior será a capacidade do espaço vetorial detectá-las e capturá-las.

Já nos *embeddings* para tweets, foram utilizados aproximadamente 3 milhões de postagens, o que é um número relativamente pequeno, se compararmos com os conjuntos de dados reportados em LI et al. (2017b) ou PENNINGTON; SOCHER; MANNING (2014), onde foram utilizados aproximadamente 2 bilhões de tweets. Mais uma vez, quanto maior for a base de dados utilizada para treinar os *embeddings*, maior será a capacidade do

algoritmo aprender as diferentes relações semânticas existentes e, portanto, maior será a qualidade do espaço vetorial gerado.

Ao realizarmos a escolha dos espaços vetoriais para tarefas que envolvam esse tipo de Esquema de Pesos, mostrou-se vantajoso empregar *embeddings* pré-treinados, que utilizaram uma base de dados para treinamento suficientemente grande, do que utilizar *embeddings* específicos para a tarefa, treinados em conjuntos de texto de tamanho bem reduzido (como foi o caso apresentado nesse trabalho).

7 CONCLUSÃO

Nesse trabalho, foi apresentado um método para a recuperação de tweets e notícias mais similares, do ponto de vista semântico. O método, originalmente aplicado para recuperar páginas da Wikipedia em inglês e em francês, se mostrou robusto o suficiente para ser aplicado no contexto de encontrar tweets e notícias que tratam sobre um mesmo assunto.

Três aspectos principais da abordagem puderam ser avaliados: o estabelecimento de relações semânticas multi-línguas, através da utilização do alinhamento de espaços vetoriais; a hipótese de podermos tratar tweets e notícias como idiomas diferentes, e a utilização da Word Mover's Distance, que é capaz de incorporar a similaridade semântica durante o cálculo das similaridades.

O primeiro aspecto se mostrou satisfatório: através do alinhamento de espaços vetoriais, conseguimos estabelecer relações semânticas multi-línguas com sucesso, como observado durante a reprodução dos experimentos envolvendo as páginas da Wikipedia em inglês e em francês.

O segundo não se mostrou tão vantajoso, uma vez que, embora essa escolha tenha apresentado um ganho, foi somente marginal, o que sugere a necessidade de maiores estudos nesse sentido. Além disso, quando utilizamos o mesmo espaço vetorial para tweets e notícias, obtivemos o melhor resultado para a tarefa.

A WMD também também foi um sucesso na abordagem, conseguindo capturar as informações semânticas presentes nos documentos e apresentando um ganho de 15% na precisão em relação à medida de similaridade utilizada no artigo original. Diante dos resultados apresentados, é válido estendermos essa combinação de uma distância que consegue incorporar o significado das palavras de *embeddings* multi-línguas para outras tarefas, como a recuperação de tweets e notícias em idiomas diferentes, por exemplo.

Conseguimos obter semântica por meio da utilização das representações vetoriais de palavras, que capturam de maneira eficiente as mais diferentes relações sintáticas e semânticas entre as palavras de um mesmo idioma, aliadas ao alinhamento de espaços vetoriais, que nos permite aumentar a quantidade de relações de significado representadas por meio do estabelecimento de relações entre palavras de diferentes idiomas.

O bom resultado do WMD frente ao método clássico evidencia a importância da utilização da representação vetorial que consegue capturar estruturas semânticas, além fornecer uma melhor interpretação sobre a similaridade de sentido entre documentos.

Dessa maneira, podemos conjecturar que a metodologia utilizada é adaptável para diferentes contextos de recuperação de informação, uma vez que não realizamos nenhum tipo de suposição em relação à natureza dos nossos documentos. São necessários mais experimentos, aplicados a contextos diferentes, para obtermos uma conclusão mais precisa

sobre a adaptabilidade da solução proposta.

Sendo assim, a abordagem apresentada fornece uma maneira de medir similaridade entre documentos que efetivamente considera a informação semântica fornecida pelos espaços vetoriais de palavras.

7.1 TRABALHOS FUTUROS

Existem algumas possíveis extensões deste trabalho a serem consideradas: a primeira delas, e a mais natural no nosso ponto de vista, é testarmos a abordagens em outros contextos. Tentar relacionar páginas da Wikipedia em outros idiomas, como o inglês e o português, estendendo o trabalho original (BALIKAS et al., 2018), ou ainda tentar recomendar tweets e notícias em idiomas diferentes. Essa é uma possibilidade interessante se consideramos que, com a alta disseminação de informação pela internet, através de redes sociais e portais de notícias, é natural que indivíduos de países que falam línguas diferentes reajam a pronunciamentos de figuras importantes do cenário mundial, seja na música, esportes, política e etc.

Também é válida uma maior investigação em relação ao impacto da base de dados sobre a qual os espaços vetoriais são treinados para o desempenho da tarefa. Pode-se verificar o desempenho da abordagem utilizando *embeddings* específicos para tweets e notícias treinados em bases de dados maiores e avaliar, sob essas circunstâncias, se é vantajoso considerarmos tweets e notícias como "línguas" diferentes.

Cabe ainda uma análise dos tempos de execução do método para diferentes tamanhos de bases de dados, uma vez que o custo computacional da abordagem proposta é elevado, não podendo ser neglicenciado.

Outras possibilidades dizem respeito à eficiência computacional do método apresentado. A utilização da WMD para relacionarmos tweets e notícias mostrou resultados superiores àquelas obtidas por distâncias como similaridade cosseno, porém o custo computacional dessa melhoria não pode ser negligenciado, uma vez que o EMD possui complexidade $\mathcal{O}(n^3 \log n)$, onde n é o tamanho do vocabulário dos documentos (PELE; WERMAN, 2009). Essa complexidade pode ser proibitiva para uma aplicação onde temos um grande número de documentos a serem relacionados, fazendo-se necessária uma análise dos tempos de execução do método.

Uma alternativa para esse problema é incorporação de uma penalidade à função objetivo do EMD, como foi feito em CUTURI (2013), para que se torne viável a utilização de algoritmos mais eficientes do ponto de vista computacional.

Outra alternativa, proposta em KUSNER et al. (2015), seria reduzir o número de possíveis candidatos a documentos de fato relevantes, utilizando as representações vetoriais das palavras que compõem os documentos como critério de seleção.

Uma outra possibilidade seria realizarmos algum agrupamento preliminar em nossos

documentos, com o intuito de filtrar os documentos com os quais precisamos calcular a similaridade, utilizando apenas aqueles que pertencem a uma mesma categoria.

REFERÊNCIAS

ALSAEDI, N.; BURNAP, P.; RANA, O. Automatic Summarization of Real World Events Using Twitter. In: **International Conference on Weblogs and Social Media**. [S.l.: s.n.], 2016. ISBN 9781577357582.

ARJOVSKY, M.; CHINTALA, S.; BOTTOU, L. Wasserstein generative adversarial networks. In: PRECUP, D.; TEH, Y. W. (Ed.). **Proceedings of the 34th International Conference on Machine Learning**. International Convention Centre, Sydney, Australia: PMLR, 2017. (Proceedings of Machine Learning Research, v. 70), p. 214–223. Disponível em: <http://proceedings.mlr.press/v70/arjovsky17a.html>.

ARTETXE, M.; LABAKA, G.; AGIRRE, E. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In: **EMNLP**. [S.l.]: The Association for Computational Linguistics, 2016. p. 2289–2294.

ASUR, S.; HUBERMAN, B. A. Predicting the future with social media. In: **Proceedings - 2010 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2010**. [S.l.: s.n.], 2010. ISBN 9780769541914. ISSN 03062619.

BALASUBRAMANYAN, R.; ROUTLEDGE, B. R.; SMITH, N. A. **From Tweets to Polls : Linking Text Sentiment to Public Opinion Time Series**. 2010.

BALIKAS, G. et al. Cross-lingual document retrieval using regularized wasserstein distance. In: **ECIR**. [S.l.]: Springer, 2018. (Lecture Notes in Computer Science, v. 10772), p. 398–410.

BENGIO, Y.; DUCHARME, R.; VINCENT, P. A neural probabilistic language model. In: LEEN, T. K.; DIETTERICH, T. G.; TRESP, V. (Ed.). **Advances in Neural Information Processing Systems 13**. MIT Press, 2003. p. 932–938. Disponível em: <http://papers.nips.cc/paper/1839-a-neural-probabilistic-language-model.pdf>.

BOJANOWSKI, P. et al. Enriching word vectors with subword information. **Transactions of the Association for Computational Linguistics**, v. 5, p. 135–146, 2017. ISSN 2307-387X.

BOLLEN, J.; MAO, H.; ZENG, X. Twitter mood predicts the stock market. **Journal of Computational Science**, v. 2, n. 1, p. 1 – 8, 2011. ISSN 1877-7503. Disponível em: <http://www.sciencedirect.com/science/article/pii/S187775031100007X>.

CAO, H. et al. A distribution-based model to learn bilingual word embeddings. In: CALZOLARI, N.; MATSUMOTO, Y.; PRASAD, R. (Ed.). **COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan**. ACL, 2016. p. 1818–1827. ISBN 978-4-87974-702-0. Disponível em: <http://aclweb.org/anthology/C/C16/C16-1171.pdf>.

CHEN, H. Collaborative systems: solving the vocabulary problem. **Computer**, v. 27, n. 5, p. 58–66, May 1994. ISSN 0018-9162.

CHONG, F.; CHUA, T. Automatic Summarization of Events From Social Media. In: **Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media Automatic**. [S.l.: s.n.], 2013. ISBN 978-1-57735-610-3.

CISSE, M. et al. Parseval networks: Improving robustness to adversarial examples. In: PRECUP, D.; TEH, Y. W. (Ed.). **Proceedings of the 34th International Conference on Machine Learning**. International Convention Centre, Sydney, Australia: PMLR, 2017. (Proceedings of Machine Learning Research, v. 70), p. 854–863. Disponível em: <http://proceedings.mlr.press/v70/cisse17a.html>.

CONNEAU, A. et al. Word translation without parallel data. **arXiv preprint arXiv:1710.04087**, 2017.

COOTES, T. F. et al. Active shape models—their training and application. **Comput. Vis. Image Underst.**, Elsevier Science Inc., New York, NY, USA, v. 61, n. 1, p. 38–59, jan. 1995. ISSN 1077-3142. Disponível em: <http://dx.doi.org/10.1006/cviu.1995.1004>.

CUTURI, M. Sinkhorn Distances: Lightspeed Computation of Optimal Transportation Distances. **Nucleic acids symposium series**, n. 22, p. 49–50, jun 2013. ISSN 0261-3166. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/1714571http://arxiv.org/abs/1306.0895>.

FUGLEDE, B.; TOPSOE, F. Jensen-Shannon divergence and Hilbert space embedding. In: **IEEE International Symposium on Information Theory**. [S.l.: s.n.], 2004. p. 31–31.

FURNAS, G. W. et al. The vocabulary problem in human-system communication. **Commun. ACM**, ACM, New York, NY, USA, v. 30, n. 11, p. 964–971, nov. 1987. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/32206.32212>.

GOLUB, G. H.; REINSCH, C. Singular value decomposition and least squares solutions. **Numer. Math.**, Springer-Verlag New York, Inc., Secaucus, NJ, USA, v. 14, n. 5, p. 403–420, abr. 1970. ISSN 0029-599X. Disponível em: <http://dx.doi.org/10.1007/BF02163027>.

GOODFELLOW, I. J. et al. Generative adversarial networks. **CoRR**, abs/1406.2661, 2014. Disponível em: <http://arxiv.org/abs/1406.2661>.

GOWER, J.; DIJKSTERHUIS, G. Procrustes problems. Vol. 30, 01 2005.

GUO, W. et al. Linking Tweets to News : A Framework to Enrich Short Text Data in Social Media. **Acl**, 2013.

HARRIS, Z. Distributional structure. **Word**, v. 10, n. 23, p. 146–162, 1954.

HOWARD, J.; RUDER, S. Fine-tuned language models for text classification. **CoRR**, abs/1801.06146, 2018.

KULLBACK, S.; LEIBLER, R. A. On information and sufficiency. **Ann. Math. Statist.**, v. 22, n. 1, p. 79–86, 1951.

KUSNER, M. J. et al. From word embeddings to document distances. In: **ICML**. [S.l.: s.n.], 2015.

LAMPLE, G. et al. Neural architectures for named entity recognition. In: **Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**. Association for Computational Linguistics, 2016. p. 260–270. Disponível em: <http://www.aclweb.org/anthology/N16-1030>.

LI, Q. et al. Data sets: Word embeddings learned from tweets and general data. **CoRR**, abs/1708.03994, 2017.

LI, Q. et al. Data sets: Word embeddings learned from tweets and general data. In: **ICWSM**. [S.l.: s.n.], 2017.

LIU, X. et al. Reuters tracer: Toward automated news production using large scale social media data. In: **Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017**. [S.l.: s.n.], 2018. ISBN 9781538627143.

MARTINEZ-ALVAREZ, M. et al. Report on the 1st international workshop on recent trends in news information retrieval (newsir16). **SIGIR Forum**, v. 50, n. 1, p. 58–67, 2016.

MIKOLOV, T. et al. Efficient estimation of word representations in vector space. **CoRR**, abs/1301.3781, 2013. Disponível em: <http://dblp.uni-trier.de/db/journals/corr/corr1301.html#abs-1301-3781>.

MIKOLOV, T. et al. Distributed representations of words and phrases and their compositionality. In: BURGES, C. J. C. et al. (Ed.). **Advances in Neural Information Processing Systems 26**. Curran Associates, Inc., 2013. p. 3111–3119. Disponível em: <http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf>.

MOGADALA, A.; JUNG, D.; RETTINGER, A. Linking Tweets with Monolingual and Cross-Lingual News using Transformed Word Embeddings. oct 2017. Disponível em: <http://arxiv.org/abs/1710.09137>.

MORIN, F.; BENGIO, Y. Hierarchical probabilistic neural network language model. In: **AISTATS**. [S.l.]: Society for Artificial Intelligence and Statistics, 2005.

NICULAE, V. Word mover's distance in python. 2015. Disponível em: <http://vene.ro/blog/word-movers-distance-in-python.html>.

PELE, O.; WERMAN, M. Fast and robust earth mover's distances. In: **ICCV**. [S.l.]: IEEE Computer Society, 2009. p. 460–467.

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

PLANK, B.; SØGAARD, A.; GOLDBERG, Y. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In: **Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)**. Association for Computational Linguistics, 2016. p. 412–418. Disponível em: <http://www.aclweb.org/anthology/P16-2067>.

RAPP, R. Identifying word translations in non-parallel texts. In: **Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics**. Stroudsburg, PA, USA: Association for Computational Linguistics, 1995. (ACL '95), p. 320–322. Disponível em: <https://doi.org/10.3115/981658.981709>.

RUBNER, Y.; TOMASI, C.; GUIBAS, L. J. A metric for distributions with applications to image databases. In: **Proceedings of the Sixth International Conference on Computer Vision**. Washington, DC, USA: IEEE Computer Society, 1998. (ICCV '98), p. 59–. ISBN 81-7319-221-9. Disponível em: <http://dl.acm.org/citation.cfm?id=938978.939133>.

SAKAKI, T.; OKAZAKI, M.; MATSUO, Y. Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors. In: **World Wide Web Conference**. [S.l.: s.n.], 2010. ISBN 978-1-60558-799-8. ISSN 1605587990.

SALTON, G.; WONG, A.; YANG, C. S. A vector space model for automatic indexing. **Commun. ACM**, ACM, New York, NY, USA, v. 18, n. 11, p. 613–620, nov. 1975. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/361219.361220>.

SHWARTZ, V. Representing words. 2006. Disponível em: <http://veredshwartz.blogspot.com/2016/01/representing-words.html>.

SMITH, S. L. et al. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. **CoRR**, abs/1702.03859, 2017. Disponível em: <http://arxiv.org/abs/1702.03859>.

SOLUTIONS, D. Word embeddings for natural language processing. 2016. Disponível em: <http://www.deep-solutions.net/blog/WordEmbeddings.html>.

SPEER, R.; LOWRY-DUDA, J. Conceptnet at semeval-2017 task 2: Extending word embeddings with multilingual relational knowledge. **CoRR**, abs/1704.03560, 2017.

SUAREZ, A. et al. A data collection for evaluating the retrieval of related tweets to news articles. In: PASI, G. et al. (Ed.). **Advances in Information Retrieval**. Cham: Springer International Publishing, 2018. p. 780–786. ISBN 978-3-319-76941-7.

SUAREZ DYAA ALBAKOUR, D. C. M. M. A.; ESQUIVEL, J. A data collection for evaluating the retrieval of related tweets to news articles. In: **40th European Conference on Information Retrieval Research (ECIR 2018), Grenoble, France, March, 2018.** [S.l.: s.n.], 2018.

TENSORFLOW. Vector representations of words. 2018. Disponível em: <https://www.tensorflow.org/tutorials/representation/word2vec>.

TOMASI, C.; KANADE, T. Shape and motion from image streams under orthography: A factorization method. **Int. J. Comput. Vision**, Kluwer Academic Publishers, Hingham, MA, USA, v. 9, n. 2, p. 137–154, nov. 1992. ISSN 0920-5691. Disponível em: <http://dx.doi.org/10.1007/BF00129684>.

TSAGKIAS, M.; RIJKE, M. de; WEERKAMP, W. Linking online news and social media. In: **Proceedings of the fourth ACM international conference on Web search and data mining - WSDM '11**. [S.l.: s.n.], 2011. ISBN 9781450304931. ISSN 10569219.

VILLANI, C. **Optimal Transport: Old and New**. 2009. ed. Springer, 2008. Hardcover. (Grundlehren der mathematischen Wissenschaften). ISBN 3540710493. Disponível em: <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20{&}path=ASIN/3540710>.

WENG, L. From gan to wgan. 2017. Disponível em: <https://lilianweng.github.io/lil-log/2017/08/20/from-GAN-to-WGAN.html>.

YU, X.; VU, N. T. Character composition model with convolutional neural networks for dependency parsing on morphologically rich languages. **CoRR**, abs/1705.10814, 2017. Disponível em: <http://arxiv.org/abs/1705.10814>.

ZHANG, M. et al. Earth mover's distance minimization for unsupervised bilingual lexicon induction. In: **EMNLP**. [S.l.]: Association for Computational Linguistics, 2017. p. 1934–1945.

ZHAO, W. X. et al. Comparing Twitter and Traditional Media Using Topic Models. In: CLOUGH, P. et al. (Ed.). **Advances in Information Retrieval**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 338–349. ISBN 978-3-642-20161-5.