UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO - ICOMP
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

MODELOS COMPOSICIONAIS: ANÁLISE E APLICAÇÃO EM PREVISÕES NO MERCADO DE AÇÕES

DIEGO FALCÃO DE SOUZA

MANAUS/AM
2017

UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO - ICOMP
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

MODELOS COMPOSICIONAIS: ANÁLISE E APLICAÇÃO EM PREVISÕES NO MERCADO DE AÇÕES

DIEGO FALCÃO DE SOUZA

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Informática do Instituto de Computação da Universidade Federal do Amazonas como requisito para a obtenção do título de Mestre em Informática.

Orientador: Dr. Edleno Silva de Moura
Coorientador: Dr. Moisés Gomes de Carvalho

MANAUS/AM
2017

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

Souza, Diego Falcão de
S729m Modelos Composicionais: Análise e Aplicação em Previsões no Mercado de Ações / Diego Falcão de Souza. 2017
44 f.: il. color; 31 cm.

Orientador: Dr. Edleno Silva de Moura
Coorientador: Dr. Moisés Gomes de Carvalho
Dissertação (Mestrado em Informática) - Universidade Federal do Amazonas.

1. Representação distribuída de palavras. 2. word embedding. 3. modelos composicionais. 4. aprendizado de máquina. 5. previsão de preços no mercado de ações. I. Moura, Dr. Edleno Silva de II. Universidade Federal do Amazonas III. Título

FOLHA DE APROVAÇÃO

"Modelos Composicionais: Análise e Aplicação em Previsões no Mercado de Ações"

DIEGO FALCÃO DE SOUZA

Dissertação de Mestrado defendida e aprovada pela banca examinadora constituída pelos Professores:

Prof. Edleno Silva de Moura - PRESIDENTE

Prof. Marco Antonio Pinheiro de Cristo - MEMBRO INTERNO

Prof. David Braga Fernandes de Oliveira - MEMBRO EXTERNO

Prof. Moisés Gomes de Carvalho - MEMBRO EXTERNO

Prof. André Luiz da Costa Carvalho - MEMBRO EXTERNO

Manaus, 10 de Julho de 2017

RESUMO

Dentre as várias técnicas de representação textual existentes na literatura, a representação distribuída de palavras (*word embedding*) vem se destacando ultimamente em várias tarefas de processamento de linguagem natural através de suas representações baseadas em vetores densos de d dimensões que são capazes de capturar informações semânticas e sintáticas das palavras. Desta forma, espera-se que as palavras com semelhanças sintáticas e semânticas estejam mais próximas umas das outras no espaço vetorial. No entanto, enquanto essa representação tem se mostrado eficaz para palavras isoladas, não há um consenso na literatura em relação à melhor forma de representar estruturas mais complexas, como frases e orações. A tendência dos últimos anos é a utilização dos modelos composicionais que representam essas estruturas complexas através da composição das representações de suas estruturas constituintes utilizando alguma função de combinação. Entretanto, sabe-se que os resultados obtidos pelos modelos composicionais dependem diretamente do domínio em que são aplicados. Nesse trabalho, nós analisamos diversos modelos de composição aplicados ao domínio de previsão de preços no mercado de ações com o objetivo de identificar qual desses modelos melhor representa os títulos de notícias financeiras para diversos métodos de aprendizado de máquina com o intuito de prever a polaridade do índice da bolsa de valore S & P 500.

Palavras-chave: Representação distribuída de palavras, *word embedding*, modelos composicionais, aprendizado de máquina, previsão de preços no mercado de ações.

ABSTRACT

Among several textual representation techniques in the literature, the distributed representation of words is standing out recently in many tasks of Natural Language Processing through its representations based on dense vectors of d dimensions that can capture syntactic and semantic information of the words. Therefore, it's expected that similar words regarding to syntactic and sematic are closer of each other in the vector space. However, while this representation is becoming effective to isolated words, there isn't a consensus in the literature regarding to the best way to represent more complex structures, such as phrases and sentences. The trend of recent years is the use of compositional models that represents these complex structures through the composition of the representations of its constituent structures using some combination function. However, it's known that the obtained results by this technique depends directly of the domain in which they are applied. In this work, we analyzed several compositional models applied to the domain of stock price prediction in order to identify which of these models better represent the financial news title for various machine learning methods to predict the index polarity of the S & P 500 stock exchange.

Keywords: Distributed representation of words, *word embedding*, compositional models, Machine learning, stock price prediction.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1. INTRODUÇÃO

Com o advento da Internet e das bibliotecas digitais, uma quantidade massiva de dados não estruturados tornou-se disponível na WEB. Esses dados, muitas vezes em formato textual, se representados e processados da forma correta, podem ser transformados em informações úteis para computadores resolverem problemas em várias áreas do conhecimento. A área de pesquisa e aplicação que explora como computadores podem ser usados para entender e manipular texto em linguagem natural para desempenhar tarefas interessantes é conhecida como PLN (Processamento de Linguagem Natural) (LIDDY, 2001).

As aplicações de PLN incluem uma série de campos de estudo, tais como tradução, processamento e sumarização de texto em linguagem natural, recuperação de informação, análise de sentimento, reconhecimento de fala, e etc. (LIDDY, 2001). Para essas várias aplicações, uma representação textual de tamanho fixo para textos de tamanho variável é importante (LEV; KLEIN; WOLF, 2015).

Diversas representações textuais têm sido utilizadas na literatura como *bag-of-words*, sintagmas nominais e entidades nomeadas. No entanto, essas representações são consideradas rasas, ou seja, não são capazes de capturar informações estruturadas sobre a relação entre as entidades existentes no texto (DING et al., 2014).

Segundo os trabalhos de (DING et al., 2014) e (DING et al., 2015), uma forma eficaz de representação textual são os eventos estruturados, pois eles capturam o que as técnicas supracitadas são incapazes de capturar: os eventos-chave embutidos no texto. No entanto, segundo os autores, essa representação é extremamente esparsa, o que limita potencialmente a capacidade de previsão dos modelos de aprendizado de máquina que a utilizam.

Ultimamente, muitos estudos em PLN têm adotado a representação distribuída de palavras no espaço vetorial (*word embedding*) proposta por (RUMELHART; HINTON; WILLIAMS, 1986) para a representação de palavras, reconhecimento de entidades, tradução, entre outros (LE; MIKOLOV, 2014). Essa representação ajuda algoritmos de aprendizado de máquina a atingirem melhores resultados em tarefas de PLN, através do agrupamento de palavras similares (MIKOLOV et al., 2000). Essa representação é capaz de capturar informações semânticas e sintáticas das palavras, de modo que palavras semanticamente similares sejam mapeadas para posições similares no espaço vetorial (LE; MIKOLOV, 2014).

Enquanto muitas pesquisas têm sido direcionadas para formas mais eficazes de construir representações para palavras individuais, não há um consenso em relação à melhor representação para estruturas mais complexas, como frases e orações (BLACOE; LAPATA, 2012).

Recentemente, diversas propostas têm sido elaboradas para computar o significado da combinação de palavras no espaço vetorial para estruturas complexas, motivadas pela popularidade da representação distribuída de palavras e sua aplicação em tarefas que exigem o entendimento de frases e sentenças completas (BLACOE; LAPATA, 2012). Essas propostas são chamadas na literatura de composição de vetores ou modelos composicionais e estão se transformando em uma tendência, recebendo muita atenção nos últimos anos (LE; MIKOLOV, 2014).

Segundo (MITCHELL; LAPATA, 2008), os modelos de composição possuem a ideia central de que o significado de uma expressão complexa é determinado pelo significado de suas expressões constituintes e pela regra utilizada para combiná-las. Todavia, segundo os autores, não há na literatura um consenso em relação a qual modelo de composição melhor se aplica a um determinado domínio específico.

Nesse trabalho, nós analisamos diversos modelos de composição aplicados ao domínio de previsão de preços no mercado de ações com o objetivo de identificar qual desses modelos melhor representa os títulos das notícias financeiras para os diversos métodos de aprendizado de máquina escolhidos.

A previsão do comportamento dos preços a curto, médio e longo prazos no mercado de ações é uma tarefa complexa tanto para modelos matemáticos quanto para especialistas humanos, visto que esse mercado é caracterizado por incerteza (AGRAWAL; CHOURASIA; MITTRA, 2013), volatilidade e dependente de diversos fatores que o influenciam tanto positiva quanto negativamente, como a política, a economia e a expectativa dos investidores (HASSAN; NATH, 2005). Entretanto, diversas soluções promissoras têm sido desenvolvidas ao longo dos anos com o objetivo de resolver esse problema, como em (AGRAWAL; CHOURASIA; MITTRA, 2013), (ATSALAKIS; VALAVANIS, 2009), (LUSS; D'ASPREMONT, 2009), (DING et al., 2014) e (DING et al., 2015). Essas soluções incluem a escolha dos tipos de dados que servirão de entrada para o método de previsão, as técnicas de representação desses dados e o método de previsão em si.

Segundo os trabalhos de (LUSS; D'ASPREMONT, 2009; MITTERMAYER;

KNOLMAYER, 2006; TETLOCK, 2007; XIE; PASSONNEAU; WU, 2013), as notícias financeiras são uma importante fonte para a previsão no mercado de ações, pois afetam drasticamente os preços. Essa evidência parte da premissa de que as notícias afetam as decisões humanas e a volatilidade dos preços das ações é influenciada por humanos. Logo, as notícias deveriam influenciar o mercado de ações (DING et al., 2014).

Considerando o exposto, temos a seguinte questão de pesquisa:

Considerando diferentes algoritmos de aprendizado de máquina, como notícias financeiras deveriam ser representadas para ajudar a prever o comportamento dos preços futuros de ações no mercado financeiro?

A busca pela resposta para esta questão definiu o objetivo deste trabalho, descrito a seguir.

1.1. OBJETIVO

1.1.1. OBJETIVO GERAL

Analisar diversos modelos de composição de conteúdos em linguagem natural, no domínio de previsão de preços no mercado de ações, de forma a identificar qual desses modelos gera a melhor representação para um conjunto das notícias financeiras.

1.1.2. OBJETIVOS ESPECÍFICOS

O objetivo geral se traduziu nos seguintes objetivos específicos:

- Selecionar os modelos composicionais que serão comparados;
- Selecionar os métodos de aprendizado de máquina que serão utilizados para medir o desempenho dos modelos composicionais;
- Medir e analisar o desempenho das representações geradas por cada modelo composicional com o objetivo de identificar qual deles possui melhor poder de representatividade, ou seja, qual deles melhor representa os textos das notícias para um método de aprendizado de máquina. Esse desempenho será medido no domínio de previsão de preços no mercado de ações através dos resultados obtidos pelos métodos de aprendizado de máquina definidos;
- Medir e analisar o desempenho dos métodos de aprendizado de máquina definidos com o objetivo de identificar qual deles obtém o melhor resultado ao utilizarem como entrada as representações geradas pelos modelos composicionais definidos.

1.2. CONTRIBUIÇÕES

As principais contribuições desse trabalho são:

- A identificação de qual modelo composicional gera as melhores representações para os eventos estruturados extraídos das notícias financeiras da base de dados utilizada;
- A identificação de qual método de aprendizado de máquina obtém os melhores resultados ao utilizar como entrada as representações geradas pelos modelos composicionais selecionados;
- A identificação de qual combinação entre modelo composicional e método de aprendizado de máquina obtém o melhor resultado;
- *Framework* para extração das notícias financeiras da base de dados utilizada, para correlacionar as notícias com os valores dos índices da bolsa S & P 500, para a geração dos eventos estruturados a partir das notícias financeiras, para a geração dos *word embeddings* a partir dos eventos estruturados e para a geração dos eventos *embedding*s com base nos modelos composicionais selecionados para esse trabalho;
- Código-fonte da Rede Neural *Feedforward* utilizada nesse trabalho.

1.3. ORGANIZAÇÃO

Esse trabalho está organizado da seguinte forma: No capítulo 2 é apresentada a fundamentação teórica para um melhor entendimento desse trabalho. No capítulo 3 são apresentados os trabalhos relacionados. No capítulo 4 é apresentado o problema e como ele foi modelado nesse trabalho. No capítulo 5 são apresentadas as bases de dados utilizadas, os experimentos, os resultados obtidos e as considerações acerca desses resultados. No capítulo 6 são apresentadas as conclusões sobre esse trabalho e os trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

Esse capítulo apresenta de forma sucinta os principais conceitos envolvidos nesse trabalho, com o objetivo de fornecer a base teórica necessária para o seu melhor entendimento. Dentre esses conceitos, estão (a) os eventos estruturados, (b) a representação distribuída ou *embedding* que é uma representação poderosa baseada em vetores para generalização, (c) os modelos composicionais que são os modelos responsáveis por criar uma representação única para estruturas complexas, isto é, estruturas compostas por mais de uma palavra e (d) os métodos de aprendizado de máquina que foram utilizados para medir o desempenho dos modelos.

2.1. EVENTOS ESTRUTURADOS

Desde a sua invenção, o texto tem sido o repositório fundamental do conhecimento e da compreensão humana. Com a invenção da impressora, do computador e com o crescimento explosivo da Web, percebemos que a quantidade de texto prontamente acessível excedeu a habilidade dos seres humanos em consumi-la.

Claramente, a compreensão automática de texto tem o potencial de ajudar, mas as tecnologias relevantes precisam estar prontas para as demandas da Web (ETZIONI et al., 2011), pois normalmente os sistemas de extração de informação aprendem um extrator para cada relação alvo a partir de exemplos de treino rotulados, o que torna essa abordagem limitada, visto que o número de relações é grande e, muitas vezes, as relações não podem ser especificadas antecipadamente (FADER; SODERLAND; ETZIONI, 2011).

Diversas representações textuais têm sido utilizadas na literatura como *bag-of-words*, sintagmas nominais e entidades nomeadas. No entanto, essas representações são consideradas rasas, ou seja, não são capazes de capturar informações estruturadas sobre a relação entre as entidades existentes no texto (DING et al., 2014).

Em (BANKO et al., 2007a), os autores introduziram um novo paradigma de extração chamado de *Open* IE, no qual o sistema extrai do *corpus* um grande conjunto de tuplas relacionais sem requerer qualquer entrada humana ou vocabulário pré-especificado (FADER; SODERLAND; ETZIONI, 2011). Por exemplo, dada a sentença $s =$ "McCain lutou duro contra Obama, mas finalmente perdeu a eleição", um sistema que utiliza o paradigma *Open* IE deve extrair de s duas triplas: $t1 =$

$(McCain, lutou\ contra, Obama)$ e $t2 = (McCain, perdeu, a\ elei\tilde{c}ao)$ (ETZIONI et al., 2011). Essas triplas ou representações geradas por esse paradigma são chamadas de eventos estruturados em (DING et al., 2014). Nesse trabalho, também adotamos essa nomenclatura.

Vários sistemas que utilizam esse paradigma foram propostos na literatura até agora, como o *TextRunner* (BANKO et al., 2007b), WOE (WELD; WU, 2010), *StatSnowBall* (ZHU et al., 2009), Ollie (MAUSAM et al., 2012) e o mais recente Open IE 4.x, utilizado por esse trabalho para extrair as triplas dos títulos das notícias financeiras e, assim, capturar a relação existente entre as entidades.

Nesse paradigma, os textos são estruturados em forma de triplas $t = (S, R, O)$, onde S é o sujeito, O é o objeto e R é a relação entre S e O (BANKO et al., 2007a). Por exemplo, para o texto "Microsoft compra o negócio de celulares da Nokia", o seguinte evento estruturado é gerado: $t = (Microsoft,\ compra,\ negócio\ de\ celulares\ da\ Nokia)$ (DING et al., 2014). Desta forma, é possível capturar a relação entre as entidades Microsoft e Nokia na sentença. No entanto, conforme dito anteriormente, essa representação é extremamente esparsa, o que limita potencialmente a capacidade de previsão dos modelos de aprendizado de máquina que a utilizam (DING et al., 2015).

2.2. REPRESENTAÇAO DISTRIBUÍDA DE PALAVRAS

Nessa forma de representação, as palavras são codificadas como vetores densos de d dimensões em um espaço vetorial, chamados de *word embeddings* (IYYER et al., 2015).

Essa representação é capaz de capturar informações semânticas e sintáticas das palavras, de modo que palavras semanticamente similares são mapeadas para posições similares no espaço vetorial (DJURIC et al., 2015). Por exemplo, as palavras "poderoso" e "forte" estão próximas umas das outras, enquanto que "poderoso" e "Paris" estão mais distantes. Além disso, é possível extrair relações de masculino/feminino, país/cidade, tempo verbal e etc. entre duas palavras (ELMING, JAKOB AND JOHANNSEN, ANDERS AND KLERKE, SIGRID AND LAPPONI, EMANUELE AND ALONSO, HECTOR MARTINEZ AND SOGAARD, 2013), conforme podemos visualizar na Figura 1. Também é possível descobrir relações através de analogias, utilizando operações algébricas simples como: $vetor("\text{rei}") - vetor("homem") + vetor("mulher") = vetor("rainha")$ (MIKOLOV; YIH; ZWEIG, 2013).

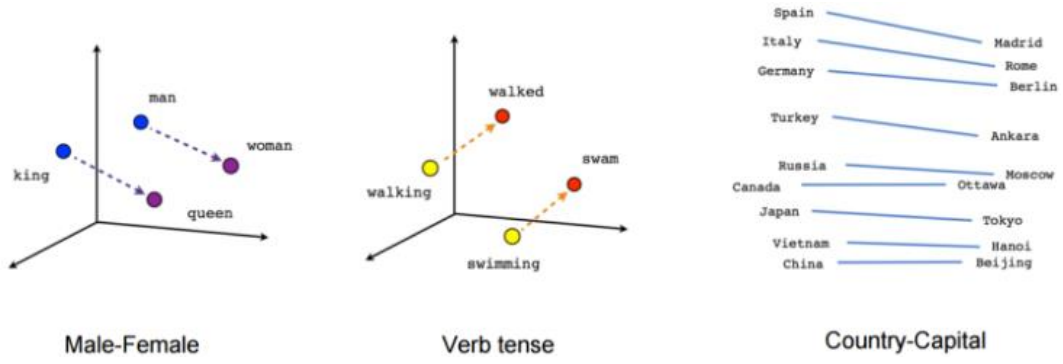

FIGURA 1. RELAÇÃO ENTRE AS REPRESENTAÇÕES DE PALAVRAS NO ESPAÇO VETORIAL[1]

Em (MIKOLOV et al., 2013a), os autores descobriram que resultados significativos podem ser obtidos através de operações simples de adição sobre os vetores. Por exemplo, pode-se descobrir a capital da Alemanha simplesmente somando-se o $vetor(\text{“}Germany\text{”})$ e o $vetor(\text{“}capital\text{”})$, obtendo assim um vetor próximo ao $vetor(\text{“}Berlin\text{”})$. Segundo os autores desse trabalho, essa composicionalidade sugere que um grau não óbvio de entendimento da linguagem pode ser obtido usando operações matemáticas simples sobre essas representações vetoriais de palavras.

Diversos modelos arquiteturais foram propostos com o objetivo de aprender representações distribuídas para as palavras. Em (MIKOLOV et al., 2013b), os autores desenvolveram um novo modelo arquitetural baseado em redes neurais que apresentou melhores resultados e minimizou a complexidade computacional em relação aos demais. Esse modelo foi chamado de word2vec e pode ser utilizado de duas formas para produzir uma representação distribuída de palavras: *continuous bag-of-words* (CBOW) ou *skip-gram*. Com o CBOW, o modelo prediz a palavra atual a partir de uma janela de palavras de contexto ao redor. A ordem das palavras de contexto não influencia a predição. Com o *skip-gram*, o modelo usa a palavra atual para prever a janela de palavras de contexto ao redor. Segundo os autores, CBOW é mais rápido enquanto o *skip-gram* é mais lento, mas faz um trabalho melhor para as palavras pouco frequentes e bases dados maiores. As duas formas podem ser visualizadas na Figura 2.

[1] Disponível em: https://www.tensorflow.org/tutorials/word2vec. Acesso em: 21/04/17

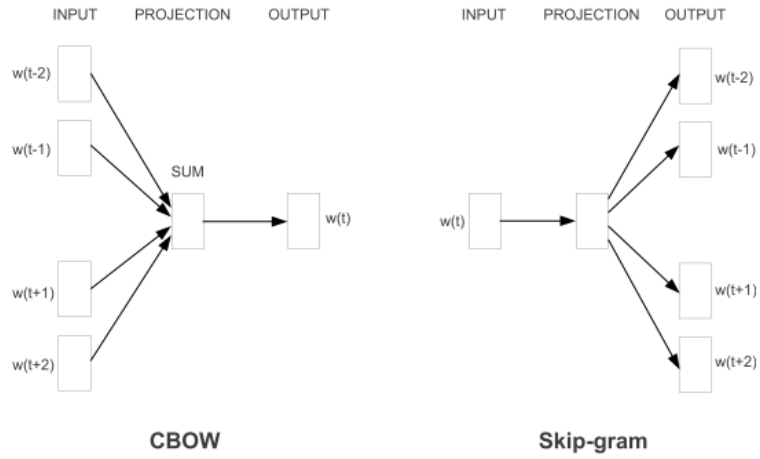

FIGURA 2. FORMAS DE UTILIZAÇÃO DO WORD2VEC (MIKOLOV ET AL., 2013B)

2.3. MODELOS COMPOSICIONAIS

Conforme supracitado, a representação distribuída de palavras para PLN representa palavras isoladas através de vetores densos de d dimensões chamados de *word embeddings*. Para aplicar essa representação para sentenças ou documentos, primeiramente devemos selecionar uma função de composição apropriada para combinar múltiplas palavras em um único vetor (IYYER et al., 2015). Essa função é chamada de modelo de composição e possui a ideia central de que o significado de uma expressão complexa é determinado pelo significado de suas expressões constituintes e pela regra utilizada para combiná-las (MITCHELL; LAPATA, 2008). No entanto, os resultados obtidos por esse modelo de composição dependem do domínio em que ele é aplicado (WIETING et al., 2016).

	animal	stable	village	gallop	jokey
horse	0	6	2	10	4
run	1	8	4	4	0

FIGURA 3. UM ESPAÇO SEMÂNTICO HIPOTÉTICO PARA AS PALAVRAS *HORSE* E *RUN*. (MITCHELL; LAPATA, 2008)

Por exemplo, levando-se em consideração modelos baseados na adição e na multiplicação, o resultado da combinação dos vetores de 5 dimensões que representam as palavras *horse* e *run* da Figura 3 seria, respectivamente:

(1) $horse + run = [1\ 14\ 6\ 14\ 4]$

(2) $horse * run = [0\ 48\ 8\ 40\ 0]$

Uma forma eficaz de resolver o problema da esparsidade dos eventos estruturados comentado na seção 2 é através da sua transformação em eventos *embeddings*, ou seja,

eventos representados através de vetores resultantes da composição de seus *embeddings* constituintes (FAN; ZHOU; ZHENG, 2017; HERMANN; BLUNSOM, 2014; LE; MIKOLOV, 2014; MIKOLOV et al., 2013a; WIETING et al., 2016). Desta forma, é possível modelar relações mais profundas entre eventos, mesmo que eles não compartilhem o mesmo sujeito, relação ou objeto (DING et al., 2015).

A maioria dos trabalhos existentes na literatura tem aplicado essas técnicas ao domínio de análise sentimento, classificação textual e recuperação da informação. No entanto, até o momento da escrita desse trabalho, apenas modelos baseados na soma, na média dos vetores e em RNT foram aplicados ao domínio de previsão de preços no mercado de ações.

2.4. MÉTODOS DE APRENDIZADO DE MÁQUINA

Nessa seção abordaremos brevemente os métodos de classificação utilizados nesse trabalho para a avaliação do desempenho das representações geradas pelos modelos composicionais.

2.4.1. KNN (*K-NEAREST NEIGHBORS*)

O KNN é um algoritmo supervisionado, ou seja, ele é um algoritmo que precisa de uma base de treino com os dados previamente rotulados com as classes que serão atribuídas posteriormente pelo modelo para as novas instâncias, pois o algoritmo primeiramente faz uma passada sobre essa base de treino e posiciona todas as instâncias rotuladas no espaço vetorial. Quando uma nova instância precisa ser classificada, o algoritmo posiciona essa nova instância no espaço vetorial e calcula a distância dela para as demais. As k instâncias mais próximas são consideradas e o rótulo mais frequente nessas instâncias é atribuído à nova instância (HARRINGTON, 2012).

2.4.2. *BAGGING*

Cada algoritmo de aprendizado de máquina possui suas forças e fraquezas, dependendo do domínio e da base de dados em que são usados. Portanto, um método que naturalmente é usado para resolver determinados problemas é justamente a combinação de vários métodos de aprendizado de máquina, conhecido na literatura por métodos *ensemble* ou meta-algoritmos. Esses métodos podem utilizar diferentes algoritmos, o mesmo algoritmo com diferentes configurações ou atribuir diferentes partes da base de dados para diferentes algoritmos (HARRINGTON, 2012).

Bootstrap aggregating ou *bagging* é uma técnica na qual os dados são retirados da base de dados original S vezes para criar S novas bases de dados. Essas bases de dados são do

mesmo tamanho que a original. Após as S bases de dados serem criadas, um algoritmo de aprendizado qualquer é aplicado a cada uma individualmente. Assim, quando uma nova instância precisa ser classificada, os S classificadores gerados são aplicados a essa instância e a classe atribuída a ela será a mais votada (HARRINGTON, 2012).

2.4.3. *ADABOOST*

Boosting é uma técnica similar ao *Bagging*. Em ambas as técnicas, o mesmo tipo de classificador é utilizado. No entanto, no *Boosting*, os diferentes classificadores são treinados sequencialmente de forma que cada novo classificador é treinado com base na performance dos classificadores que já foram treinados. Desta forma, cada novo classificador pode focar nos dados que foram previamente mal classificados pelos classificadores anteriores (HARRINGTON, 2012).

O *AdaBoost* é uma versão adaptativa do *Boosting* amplamente utilizado e desenvolvido especificamente para classificação que funciona da seguinte forma: O algoritmo inicialmente atribui pesos iguais para todas as instâncias da base de dados de treino. Após isso, ele é executado sobre os dados e os pesos são refeitos para cada instância de acordo com o resultado da execução. Os pesos das instâncias corretamente classificadas são decrementados e os pesos das instâncias incorretamente classificadas são incrementados. Esse processo de ajuste dos pesos se repete por várias iterações (WITTEN IAN H AND FRANK, EIBE AND HALL, MARK A AND PAL, 2016) até que o erro do treino chegue a 0 ou até que o número de classificadores fracos atinja um valor definido pelo usuário (HARRINGTON, 2012).

2.4.4. *RANDOM FOREST*

Assim como o *Bagging* e o *AdaBoost*, o *Random Forest* também é um meta-algoritmo. Ele é uma modificação substancial do *Bagging* que constrói uma grande coleção de árvores não relacionadas para depois tirar uma média dos resultados de cada uma. Em vários problemas, esse algoritmo possui a mesma performance que o *Boosting*, porém sendo mais simples de treinar e ajustar (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

Quando usado para classificação, o *Random Forest* obtém um voto da classe para cada árvore, e então classifica a nova instância usando o voto da maioria. Quando usado para regressão, é retirada uma média das previsões de cada árvore (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

2.4.5. ÁRVORE DE DECISÃO

A árvore de decisão utiliza uma abordagem de dividir para conquistar para aprender através de um conjunto de instâncias independentes. Primeiramente, o algoritmo seleciona um atributo para ser posto como o nó raiz da árvore. Em seguida, uma ramificação é criada para cada valor possível desse atributo. Desta forma, o algoritmo divide a base de dados em subconjuntos, um para cada valor do atributo. Esse processo se repete recursivamente até que todas as instâncias da base de dados tenham sido mapeadas. Os nós em uma árvore de decisão são utilizados para testar os valores de um atributo em particular para decidir qual ramificação seguir e os nós folha são utilizados para a classificação (WITTEN IAN H AND FRANK, EIBE AND HALL, MARK A AND PAL, 2016). Os algoritmos J48 (C4.5), *SimpleCart* e *REPTree* são exemplos de implementações de árvores de decisão.

Para classificar uma nova instância, o algoritmo caminha pela árvore previamente gerada de acordo com os valores dos atributos dessa instância até atingir um nó folha. O algoritmo então classifica essa instância de acordo com a classe referente ao nó folha atingido (WITTEN IAN H AND FRANK, EIBE AND HALL, MARK A AND PAL, 2016). Por exemplo, na Figura 4 podemos visualizar uma árvore de decisão construída para o problema de decidir se devemos ou não jogar tênis. Nesse caso, o algoritmo compara primeiramente se o clima está ensolarado, nublado ou chuvoso. Se o clima estiver nublado, o algoritmo já decide que se deve jogar tênis. Caso o clima esteja ensolarado, o algoritmo precisa saber como está a umidade antes de tomar uma decisão. Caso a umidade esteja alta, não devemos jogar tênis, caso esteja normal, devemos jogar tênis.

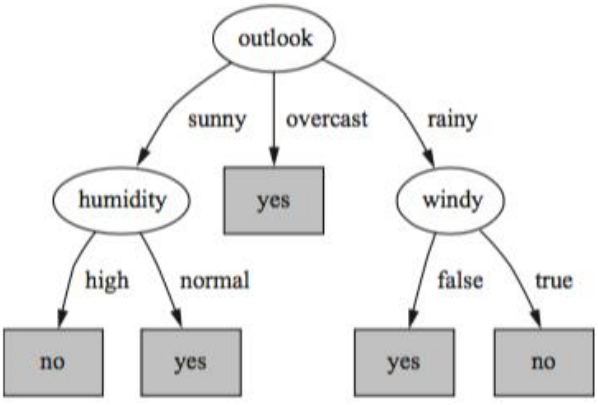

FIGURA 4. ÁRVORE DE DECISÃO PARA A TAREFA DE JOGAR TÊNIS

2.4.6. *NAIVE BAYES*

Ele é baseado na teoria Bayesiana e é considerado o mais simples algoritmo probabilístico. Ele é chamado de *naive* porque a formulação faz algumas premissas ingênuas.

Para classificar uma nova instância, nós temos uma equação para a probabilidade dessa instância pertencer à Classe 1 e uma equação para a probabilidade dessa instância pertencer à Classe 2. Portanto, para classificar, escolhemos a classe com a maior probabilidade para cada instância (HARRINGTON, 2012).

2.4.7. SVM

O SVM é considerado por algumas pessoas como o melhor classificador que não precisa ser pré-configurado para atingir bons resultados, pois podemos utilizá-lo em sua forma padrão sobre os dados e mesmo assim os resultados terão taxas baixas de erro. Esse classificador toma boas decisões para dados que estão fora do conjunto de treinamento através da criação de uma superfície de decisão que separa os dados de classes diferentes em grupos (HARRINGTON, 2012). Essa superfície de decisão é chamada de hiperplano e pode ser traçada sobre dados com n dimensões. Todas as instâncias que estiverem de um lado pertencem a uma classe e todas as instâncias que estiverem do outro lado pertencem a uma outra classe diferente.

Esse classificador pode ser aplicado a problemas não linearmente separáveis onde há sobreposição de dados, ou seja, pode ser aplicado a casos em que as classes não podem ser separadas claramente por uma superfície linear (HASTIE; TIBSHIRANI; FRIEDMAN, 2009). Um exemplo de separação linear pode ser visualizado na Figura 5.

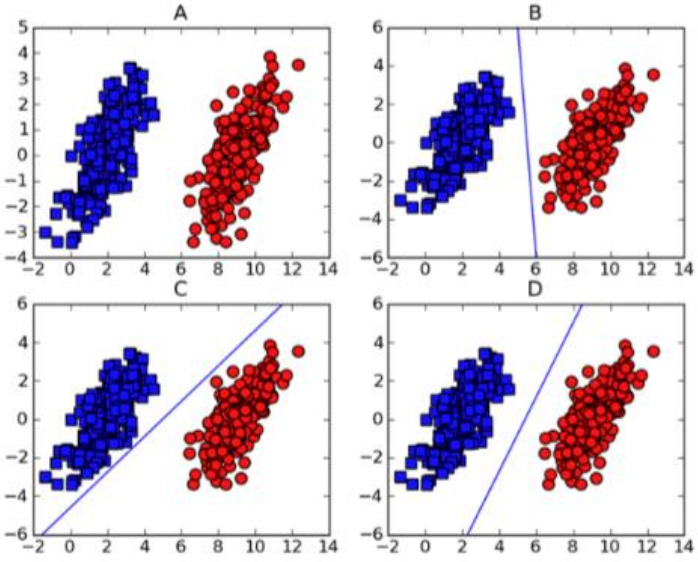

FIGURA 5. DADOS LINEARMENTE SEPARÁVEIS NO QUADRO A. QUADROS B, C E D MOSTRAM POSSÍVEIS LINHAS SEPARANDO AS DUAS CLASSES DE DADOS (HARRINGTON, 2012)

Os algoritmos SMO e libSVM são exemplos de implementações do SVM.

2.4.8. PART

Regras Se-Então são a base para algumas das linguagens de descrição de conceito mais populares usadas no aprendizado de máquina. Elas permitem que o conhecimento extraído de uma base de dados seja representado de uma forma entendida facilmente por humanos (FRANK; WITTEN, 1998).

Duas formas dominantes de implementações práticas desse aprendizado baseado em regras são o C4.5 e o RIPPER. O C4.5 gera primeiramente uma árvore de decisão grande e redundante para posteriormente executar um processo de otimização que poda algumas regras individuais com o objetivo de gerar um conjunto de regras mais preciso que o inicial. O RIPPER utiliza uma estratégia de dividir para conquistar que determina as regras base mais poderosas para a base de dados e, em seguida, separa-as dos exemplos que são cobertos por elas para então repetir o processo nos exemplos restantes. Desta forma, esse algoritmo melhora a precisão através da substituição ou da revisão de regras individuais. O PART é um algoritmo que combina os dois paradigmas supracitados para gerar um conjunto de regras compacto e preciso (FRANK; WITTEN, 1998).

2.4.9. REDE NEURAL ARTIFICIAL

Uma rede neural artificial (RNA) é baseada na rede neural biológica, onde os neurônios produzem sinais baseados nos sinais recebidos por outros neurônios e os transmitem ao longo da rede para outros neurônios (MICHALSKI; CARBONELL; MITCHELL, 2013).

As RNA podem ser vistas como um grafo ponderado direcionado no qual os neurônios artificiais são os nós e as arestas direcionadas com os pesos são as conexões entre as entradas e saídas dos neurônios. Quando esse grafo não possui laços, chamamos a RNA de *feed-forward* (JAIN; MAO; MOHIUDDIN, 1996).

O tipo mais comum de RNA *feed-forward* é o *Multilayer Perceptron*, onde os neurônios estão organizados em camadas que possuem conexões unidirecionais entre elas (JAIN; MAO; MOHIUDDIN, 1996). Na Figura 6 podemos visualizar o exemplo de uma RNA desse tipo aplicada ao problema de jogar tênis abordado anteriormente, configurada com duas camadas ocultas contendo 4 neurônios cada uma.

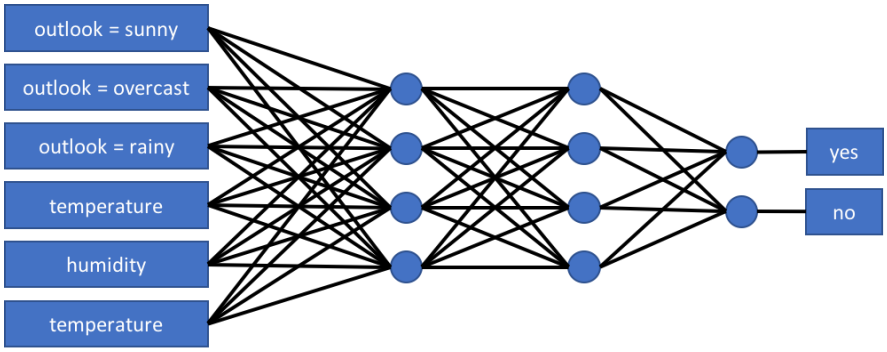

FIGURA 6. EXEMPLO DE UMA RNA COM DUAS CAMADAS OCULTAS. ADAPTADO DE (WITTEN IAN H AND FRANK, EIBE AND HALL, MARK A AND PAL, 2016)

O processo de aprendizado no contexto de uma RNA pode ser visto como um problema de atualização dos pesos das conexões entre neurônios. Desta forma, a rede é capaz de executar eficientemente uma tarefa específica. A rede geralmente deve aprender os pesos das conexões através dos padrões disponíveis na base de dados de treino, isto é, deve aprender um conjunto de regras que guiam as relações entre as entradas e as saídas de uma coleção de exemplos representativos através do ajuste dos pesos da rede iterativamente ao longo do tempo (JAIN; MAO; MOHIUDDIN, 1996).

3. TRABALHOS RELACIONADOS

Esse capítulo apresenta uma breve descrição dos trabalhos existentes na literatura relacionados a essa pesquisa. Esses trabalhos aplicaram modelos composicionais simples e complexos em diferentes domínios com o objetivo de obterem uma melhor representação textual.

A utilização de modelos composicionais para representar estruturas mais complexas que palavras isoladas é uma tendência recente e tem recebido muita atenção nos últimos anos (LE; MIKOLOV, 2014). Alguns trabalhos utilizam operações aritméticas simples em seus modelos de composição como a soma (IYYER et al., 2015), soma ponderada (YU; DREDZE, 2015), multiplicação, média (MITCHELL; LAPATA, 2010) e concatenação (BLACOE; LAPATA, 2012). Outros trabalhos utilizam modelos mais complexos que envolvem métodos de aprendizado de máquina como Redes Neurais Recursivas (SOCHER; HUANG; PENNINGTON, 2011), Redes Neurais Tensor (RNT) (DING et al., 2015), Redes Neurais Convolutivas (KALCHBRENNER; GREFENSTETTE; BLUNSOM, 2014) e LSTM (*Long Short Term Memory Network*) (TAI; SOCHER; MANNING, 2015).

3.1. MODELOS COMPOSICIONAIS SIMPLES

Em (MITCHELL; LAPATA, 2008) é feita uma comparação entre alguns modelos que utilizam operações aritméticas simples sobre os vetores como a soma, soma ponderada, multiplicação e uma combinação da multiplicação e soma para resolver o problema da multiplicação por 0. Para a tarefa de similaridade entre sentenças, os modelos que utilizaram a multiplicação e combinação obtiveram resultados estatisticamente parecidos e o modelo que utilizou a soma não obteve bons resultados.

Em (IYYER et al., 2015) é feita uma comparação entre um modelo composicional baseado na média e outro baseado na soma para a tarefa de classificação textual. Segundo os autores, os experimentos indicaram que o modelo baseado na média obteve melhores resultados em relação ao modelo baseado na soma.

Em (LIU et al., 2015) é introduzido o TWE (*Topical Word Embedding*) no qual a palavra tópico se refere a uma palavra que leva um tópico específico como contexto. A ideia básica é permitir que cada palavra tenha diferentes *embeddings* para diferentes tópicos. Por exemplo, a palavra Apple significa uma fruta no tópico alimentos e significa uma empresa de TI no tópico de tecnologia da informação (TI). Essa palavra tópico é gerada através da

concatenação do *embedding* do tópico com o *embedding* da palavra. Esse método obteve melhores resultados em relação aos demais nas tarefas de similaridade entre palavras e classificação textual.

Segundo (MITCHELL; LAPATA, 2008), o modelo mais comum de composição existente na literatura utiliza a soma vetorial, abordada em (MITCHELL; LAPATA, 2010) e (BLACOE; LAPATA, 2012). No entanto, segundo os autores, esse modelo não é capaz de representar a semântica de uma sentença, pois não leva em consideração a sintaxe e a ordem das palavras. Portanto, os autores o consideram essencialmente uma abordagem de *bag-of-words*.

Muitas entidades do mundo real são expressas por frases não composicionais, como nomes compostos de pessoas e nomes de filmes, cujo significado não pode ser composto por suas palavras constituintes (YANG et al., 2014). Mesmo assim, segundo (MITCHELL; LAPATA, 2008), modelos baseados na soma de vetores têm apresentado bons resultados em algumas tarefas específicas abordadas em (DEERWESTER et al., 1990) e (LANDAUER; DUMAIS, 1997). Essas tarefas estão mais preocupadas com a modelagem da essência de um documento em vez do significado de suas sentenças.

3.2. MODELOS COMPOSICIONAIS COMPLEXOS

Em (CHEN et al., 2013) é proposto um modelo baseado em Redes Neurais Tensor (RNT) no qual cada palavra é representada por um vetor e cada entidade é representada pela média dos vetores das palavras constituintes. Segundo os autores, essa abordagem permite o compartilhamento da força estatística entre as palavras que descrevem cada entidade. As relações entre as palavras são definidas através dos parâmetros de uma RNT, capazes de relacionar explicitamente duas palavras. Segundo esse trabalho, uma RNT provê uma forma mais poderosa de modelar informações relacionais que uma rede neural artificial padrão.

Em (DING et al., 2015) foi utilizada a abordagem de (CHEN et al., 2013) para representar notícias financeiras. Essa representação serviu como entrada para treinar uma rede neural convolutiva para prever o comportamento dos preços no mercado de ações. Esse trabalho obteve 6% de melhoria nas previsões em relação a outros trabalhos comparados, sendo considerado pelos autores o estado da arte.

Em (LE; MIKOLOV, 2014) é proposto um modelo chamado *Paragraph Vector* baseado em um algoritmo não supervisionado que aprende a representar pedaços de textos variáveis como sentenças, parágrafos e documentos através de vetores densos de tamanhos fixos. Resultados empíricos mostraram que o modelo proposto supera os modelos baseados

em *bag-of-words* e em outras técnicas de representação textual. Além disso, segundo os autores, esse trabalho foi considerado o estado da arte em 2014 para tarefas de classificação textual e análise de sentimento.

Em (WIETING et al., 2016) foi feita uma comparação entre diversos modelos composicionais baseados na média, média seguida por uma projeção linear simples e em três variantes de redes neurais recorrentes (RNR), incluindo LSTM. Segundo os autores, os modelos baseados em médias obtiveram melhores performances para as tarefas de similaridade e dedução, superando o LSTM. No entanto, para análise de sentimento, LSTM obteve melhores resultados.

Em (PALANGI et al., 2015) foi proposto um modelo composicional baseado em LSTM. Esse modelo foi comparado no domínio de recuperação da informação com vários outros modelos, dentre eles o *Paragraph Vector*. Para o domínio em questão, LSTM obteve melhores resultados, segundo os autores.

Diante dos trabalhos apresentados e de acordo com (WIETING et al., 2016), podemos concluir que os resultados dos modelos composicionais dependem diretamente do domínio em que são aplicados, podendo apresentar resultados bons em um domínio enquanto apresentam resultados ruins em outro.

Poucos trabalhos têm feito comparações entre modelos composicionais. Os méritos das diferentes abordagens são ilustrados com alguns exemplos escolhidos manualmente e os valores dos parâmetros e as avaliações estão ausentes (MITCHELL; LAPATA, 2008). Para o domínio de previsão de preços no mercado de ações, a quantidade de trabalhos existentes na literatura é ainda menor.

O presente trabalho se diferencia dos demais, pois investigamos os modelos composicionais mais promissores para o domínio de previsão de preços no mercado de ações e analisamos o impacto desses modelos em vários método de aprendizado de máquina, partindo da premissa identificada por (WIETING et al., 2016) de que modelos composicionais simples são capazes de obter melhores resultados em relação a modelos composicionais mais complexos, dependendo do domínio em questão. Segundo os autores, os modelos mais simples são competitivos, extremamente eficientes e fáceis de usar.

4. MODELAGEM

Esse capítulo descreve de forma objetiva o problema que esse trabalho buscou solucionar ao longo de toda a pesquisa, bem como a forma como ele foi modelado.

Segundo (WIETING et al., 2016), existem poucos trabalhos na literatura relacionados a representações de estruturas complexas que possam ser usadas entre domínios com a mesma facilidade e eficácia das representações distribuídas de palavras isoladas. Os autores afirmam que os resultados dessas representações dependem diretamente do domínio em que elas são aplicadas.

A maioria dos trabalhos existentes na literatura tem aplicado essas representações ao domínio de análise de sentimento, classificação textual e recuperação da informação. No entanto, até o momento da escrita desse trabalho, apenas representações baseadas na adição, média e RNT (Redes Neurais Tensor) foram aplicadas ao domínio de previsão de preços no mercado de ações, sem quaisquer justificativas para tais escolhas. Além disso, não é de nosso conhecimento que existam trabalhos na literatura que comparem diversas representações nesse domínio. Diante desse cenário, as seguintes hipóteses foram formuladas:

1. Existe um melhor modelo composicional para o domínio em questão?

2. Existe um melhor método de aprendizado de máquina para as representações geradas pelos modelos composicionais?

3. Existe uma melhor combinação de modelo composicional e método de aprendizado de máquina?

Nesse trabalho, nós comparamos diversos modelos composicionais no domínio de previsão de preços no mercado de ações, seguindo a metodologia utilizada por (DING et al., 2015), esboçada no fluxo da Figura 7. Os autores utilizaram um modelo composicional baseado em RNT para representar os eventos estruturados extraídos das notícias financeiras. Essas representações serviram de entrada para uma rede neural com o objetivo de prever se o índice da bolsa de valores S&P 500 aumentaria ou diminuiria.

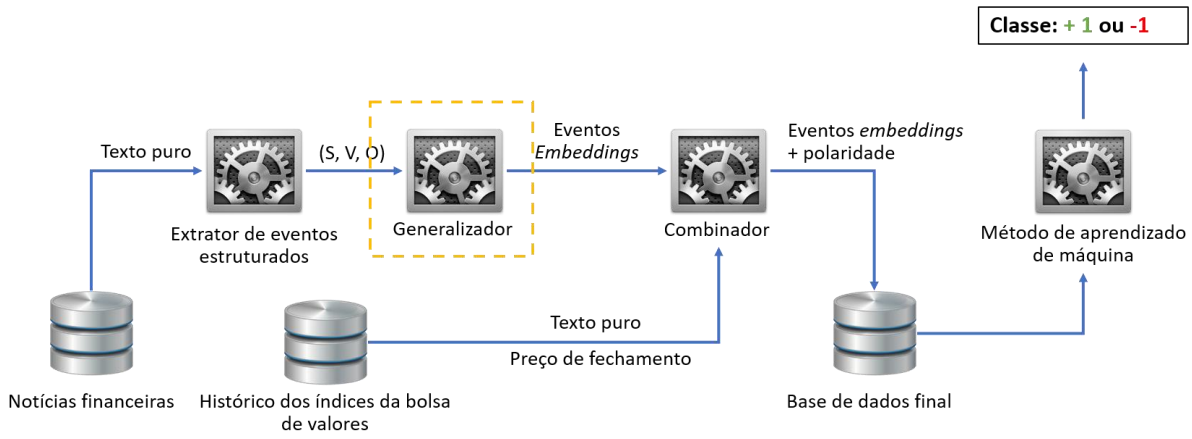

FIGURA 7. METODOLOGIA PARA AVALIAÇÃO DOS MODELOS COMPOSICIONAIS

Esse trabalho focou apenas no componente destacado na Figura 7, pois ele é o responsável por utilizar os modelos composicionais para combinar os *word embeddings* dos componentes dos eventos estruturados, gerando assim os eventos *embeddings* que representam as notícias financeiras. Os demais componentes foram implementados apenas para que a avaliação dos modelos composicionais fosse possível através dos resultados obtidos pelos métodos de aprendizado de máquina.

Existem várias formas de utilizar aprendizado de máquina para previsão no mercado de ações. Podemos prever o preço de uma ação específica, o valor ou a mudança do índice (polaridade) de uma determinada bolsa de valores ou, ainda, fazer *trading* (DING et al., 2015), isto é, executar operações automáticas de compra e venda de ações. Essas previsões podem ser de curto, médio e longo prazos.

Para medir o desempenho dos modelos composicionais selecionados para esse trabalho, vários métodos de aprendizado de máquina foram selecionados e aplicados ao problema de previsão a curto prazo da polaridade de mudança do índice da bolsa S&P 500. Desta forma, buscamos prever com base nas notícias financeiras de um dia se o índice da bolsa de valores aumentaria ou diminuiria no dia seguinte (XIE; PASSONNEAU; WU, 2013). Portanto, estamos tratando a previsão como um problema de classificação e não de série temporal, visto que consideramos a influência das notícias financeiras de apenas um dia na polaridade do índice da bolsa de valores.

Os métodos de aprendizado de máquina selecionados foram *Naive Bayes*, RNA (Rede Neural Artificial), SVM (*Support Vector Machine*), KNN (K *Nearest Neighbors*), *AdaBoost*, *Bagging*, PART, J48 e *Random Forest*, pois representam famílias diferentes de modelos.

Os modelos composicionais escolhidos foram aqueles baseados em operações aritméticas simples de adição, média, multiplicação, concatenação e o *Paragraph Vector* (LE; MIKOLOV, 2014) por se tratarem de modelos mais simples que exigem menor poder computacional que os modelos mais complexos (BLACOE; LAPATA, 2012) e que ao mesmo tempo apresentam resultados satisfatórios em vários domínios (WIETING et al., 2016).

4.1. MODELAGEM DAS NOTÍCIAS FINANCEIRAS

Apenas os títulos das notícias financeiras foram utilizados, visto que os trabalhos de (DING et al., 2014) e (RADINSKY; DAVIDOVICH; MARKOVITCH, 2012) mostram que os títulos são mais úteis para previsão que os conteúdos (DING et al., 2015).

A modelagem desses títulos foi feita em quatro etapas, conforme podemos visualizar na Figura 8. Essas etapas são descritas separadamente nas seções seguintes.

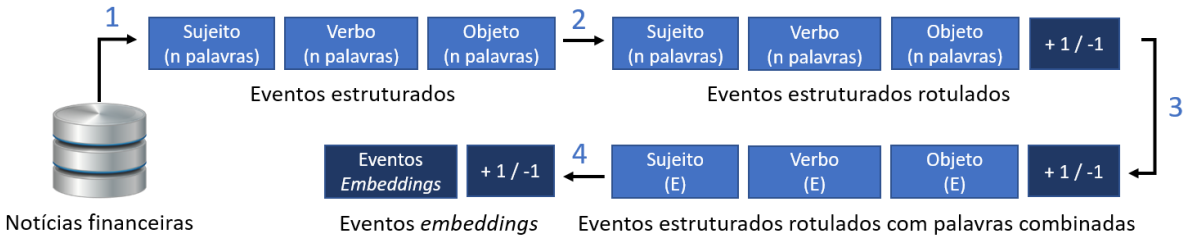

FIGURA 8. MODELAGEM DAS NOTÍCIAS FINANCEIRAS

4.1.1. TRANSFORMAÇÃO DAS NOTÍCIAS FINANCEIRAS EM EVENTOS ESTRUTURADOS

A transformação dos títulos das notícias financeiras em eventos estruturados foi feita através da tecnologia Open IE, implementada pela ferramenta Open IE 4.x[2]. Essa etapa consistiu na transformação de cada notícia em uma tripla composta por seu sujeito, verbo/relação e objeto, de acordo com o exemplo da Figura 9. As notícias que não apresentaram essa estrutura completa foram descartadas.

[2] https://github.com/knowitall/openie

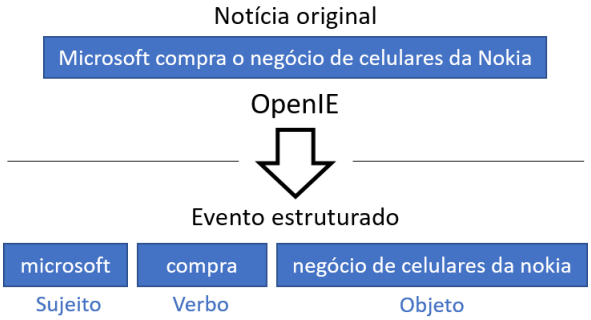

FIGURA 9. EXEMPLO DE TRANSFORMAÇÃO DE UMA NOTÍCIA EM EVENTO ESTRUTURADO

4.1.2. ROTULAGEM DOS EVENTOS ESTRUTURADOS

A rotulagem dos eventos estruturados consiste em rotulá-los de acordo com o impacto causado no índice da bolsa de valores pelas notícias financeiras que eles representam. Se a notícia financeira de um dia influencia positivamente o índice do dia seguinte da bolsa de valores, essa notícia é rotulada como +1, caso contrário, -1. Para isso, os valores do preço de fechamento do dia da notícia e o preço de fechamento do dia seguinte são comparados, conforme podemos visualizar na Figura 10. Os valores dos preços de fechamento foram extraídos do histórico de índices da bolsa S & P 500, disponível no Yahoo *Finance*[3].

Existem vários casos na base de dados em que há notícias financeiras para determinados dias, mas não há índices correspondentes a esses dias na bolsa de valores pelo fato de terem ocorrido em finais de semana ou feriados. Nesses casos, não é possível atribuir um rótulo para a notícia, pois não há como comparar os valores. Para contornar esse problema, comparamos o dia em que há índice na bolsa de valores com o próximo dia em que há índice associado.

Por exemplo, se uma determinada notícia de uma sexta-feira precisa ser rotulada e os próximos dias são sábado e domingo, o rótulo dessa notícia será atribuído através da comparação do índice da sexta-feira com o índice da segunda-feira, caso esse dia possua um índice associado. Se uma determinada notícia de um sábado precisa ser rotulada, o rótulo dessa notícia será atribuído através da comparação do índice da sexta-feira com o índice de segunda-feira, visto que não haverá índice associado ao dia de sábado.

[3] https://finance.yahoo.com

FIGURA 10. ROTULAGEM DOS EVENTOS ESTRUTURADOS

4.1.3. TRANSFORMAÇÃO DOS COMPONENTES DOS EVENTOS ESTRUTURADOS EM *EMBEDDINGS*

A transformação de cada componente da tripla de eventos estruturados em *embeddings* não foi trivial, pois uma grande parte dos eventos estruturados apresenta componentes compostos por várias palavras. Portanto, foi necessário combinar os *embeddings* dessas palavras para que cada componente tivesse uma representação única. A estratégia utilizada por esse trabalho foi a mesma adotada por (DING et al., 2015) na qual cada componente é representado pela média dos *embeddings* de suas palavras. Desta forma, se o sujeito é composto por 3 palavras, sua representação final será a média dos *embeddings* de cada uma dessas palavras, conforme podemos visualizar na Figura 11. O autor afirma que essa estratégia permite o compartilhamento da força estatística entre as palavras que descrevem cada componente.

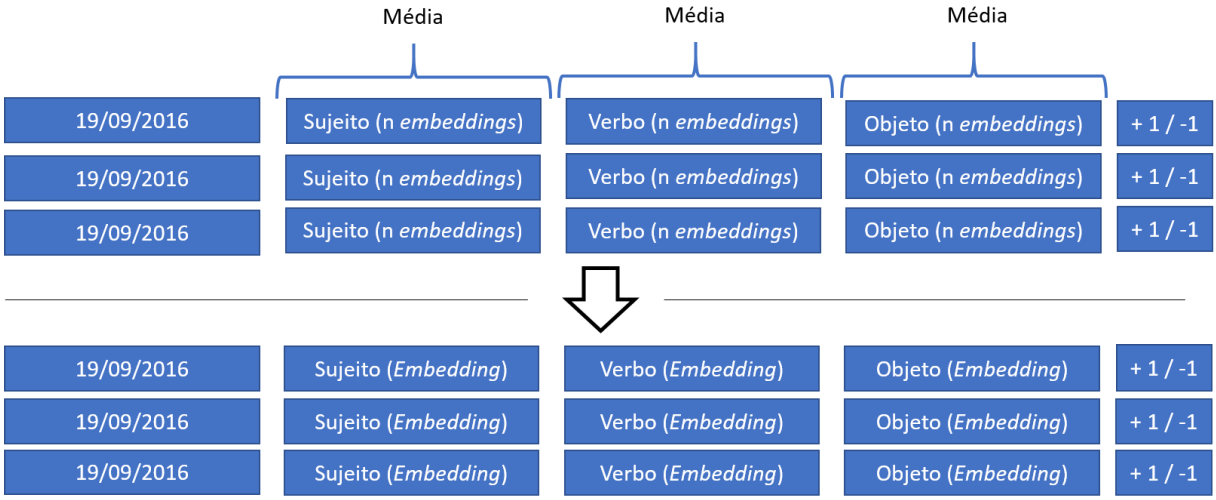

FIGURA 11. REPRESENTAÇÃO DAS NOTÍCIAS FINANCEIRAS

4.1.4. GERAÇÃO DOS EVENTOS *EMBEDDINGS*

Como podemos perceber na Figura 11, cada dia pode ser composto por várias notícias financeiras. Essas notícias foram combinadas através da média de seus *embeddings* de forma a obtermos uma representação única para cada dia, conforme podemos visualizar na Figura 12. Portanto, essa etapa consistiu nessa combinação e na geração dos eventos *embeddings* através da aplicação dos modelos composicionais sobre as triplas de eventos estruturados de cada notícia financeira, compostas por *embeddings*. Os seguintes modelos composicionais foram aplicados:

1) Adição - O modelo de composição baseado na adição combina os componentes da tripla através da soma de seus *embeddings*;

2) Multiplicação - O modelo de composição baseado em multiplicação combina os componentes da tripla através da multiplicação de seus *embeddings*;

3) Média - O modelo de composição baseado na média combina os componentes da tripla através da média de seus *embeddings;*

4) Concatenação – O modelo de composição baseado na concatenação simplesmente concatena os *embeddings* dos componentes da tripla, gerando um evento *embedding* com o triplo da dimensão dos demais;

5) *Paragraph Vector* - O modelo gera uma combinação dos *embeddings* automaticamente através de um modelo treinado previamente sobre a base de dados de notícias financeiras. O algoritmo utilizado foi o PV-DM através da ferramenta doc2vec[4] e os parâmetros utilizados para a geração do modelo foram: *size*=100, *window*=10 e *min_count*=5. Segundo (LE; MIKOLOV, 2014), PV-DM obtém melhores resultados em relação ao PV-DBOW e um valor entre 5-12 para o tamanho da janela é um bom valor.

Conforme supracitado, existem casos em que há notícias financeiras na base de dados para dias que não possuem índices associados na bolsa de valores. Para esses casos, optamos por também tirar a média dos *embeddings* das notícias. Portanto, considerando-se notícias de sexta-feira e sábado, por exemplo, o *embedding* resultante para a sexta-feira seria a média dos *embeddings* das notícias de sexta-feira e sábado e o dia de sábado seria descartado.

[4] https://radimrehurek.com/gensim/models/doc2vec.html

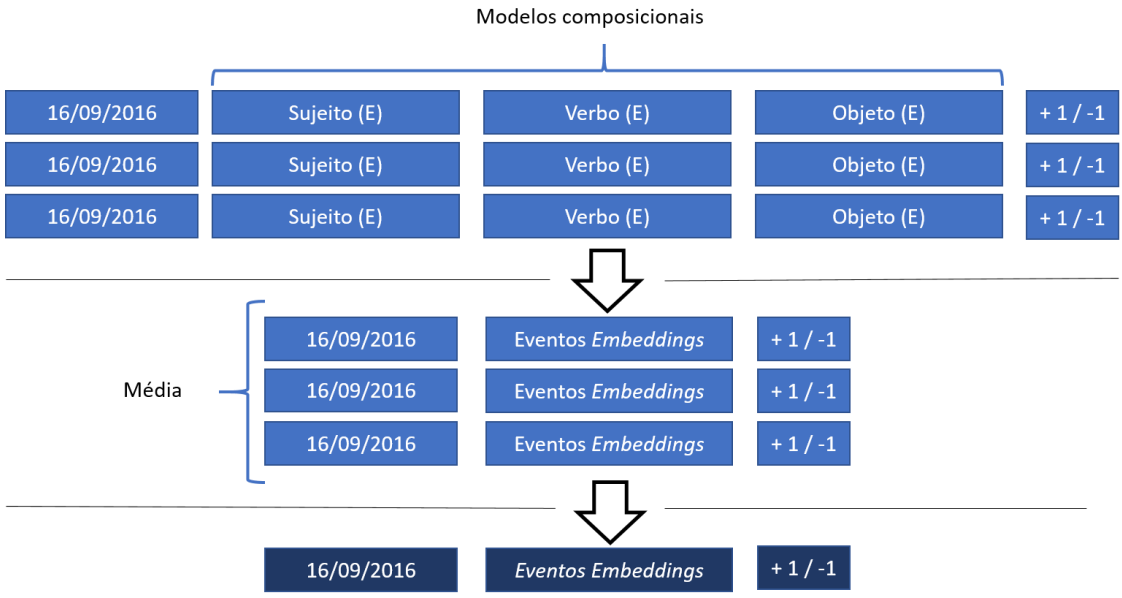

FIGURA 12. EVENTOS *EMBEDDINGS*

Logo, uma instância é composta por um evento *embedding* de *d* dimensões que representa todas as notícias de um dia e por seu respectivo rótulo, indicando se ele influenciou positiva ou negativamente o índice do dia seguinte da bolsa de valores.

Na Figura 13, podemos visualizar um exemplo da aplicação de um modelo composicional baseado na adição sobre um evento estruturado extraído de uma notícia financeira para geração de um evento *embedding*. Nota-se que o modelo faz uma soma aritmética dos valores que representam cada componente. Essa forma de cálculo considerando o *embedding* de cada componente também se repete para todos os outros modelos composicionais.

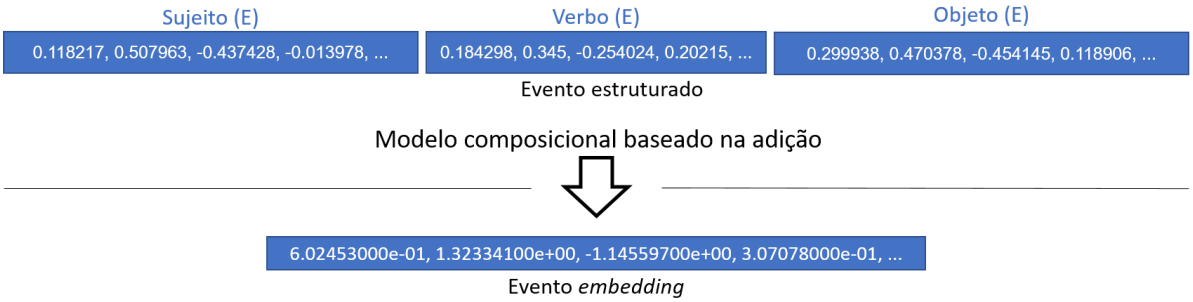

FIGURA 13. EXEMPLO DA APLICAÇÃO DO MODELO COMPOSICIONAL BASEADO NA ADIÇÃO

Essa representação resultante servirá de entrada para os métodos de aprendizado de máquina escolhidos. Desta forma, através dos resultados obtidos por esses métodos, podemos avaliar o poder de representatividade das notícias financeiras por parte de cada modelo composicional.

5. EXPERIMENTOS

Esse capítulo apresenta a metodologia utilizada para a execução dos experimentos de avaliação de desempenho dos modelos composicionais e dos métodos de aprendizado de máquina, bem como apresenta os resultados obtidos por eles.

Para a execução dos experimentos, duas bases de dados foram utilizadas. Uma delas é a base de dados de notícias financeiras que serve de entrada para a geração dos eventos *embeddings* pelos modelos composicionais e a segunda é a base de dados que serve de entrada para o treino do modelo responsável pela geração dos *embeddings* de cada palavra existente na primeira base.

5.1. BASE DE DADOS DE NOTÍCIAS FINANCEIRAS

Existem duas abordagens que podem ser adotadas ao lidar com base de dados para experimentos. A primeira delas é utilizar uma base de dados desenvolvida por terceiros que seja confiável, isto é, referenciada e adotada por outros trabalhos da literatura e a segunda é a sua construção para o domínio em questão.

Nesse trabalho, optamos pela primeira opção. No entanto, durante a fase de pesquisa, apenas uma base de dados mostrou-se disponível e confiável para o domínio. Essa base foi elaborada e utilizada por (DING et al., 2014) e (DING et al., 2015) e contém 410.234 notícias financeiras em inglês para um total de 1.944 dias, de 20/10/2006 até 26/11/2013, coletadas da empresa Bloomberg.

Para a execução dos experimentos desse trabalho, alguns pré-processamentos foram necessários sobre a base de dados, como:

- Remoção de casos genitivos, ou seja, remoção das ocorrências de apóstrofo (‘) seguido de s;

- Remoção dos caracteres especiais: exclamação, acento grave, aspas simples, aspas duplas, dois pontos, ponto e vírgula, vírgula, cerquilha, acento circunflexo, asterisco, parênteses e chaves;

- Transformação de todas as palavras para caixa baixa.

5.2. BASE DE DADOS PARA TREINO DOS *EMBEDDINGS*

Para que se obtenham melhores resultados, é recomendável que o modelo responsável pela geração dos *embeddings* seja treinado em uma base de dados que faça parte do mesmo domínio do problema, pois uma determinada palavra pode ter significados diferentes dependendo do domínio em questão. Por exemplo, no domínio de compras, a palavra bolsa refere-se a um acessório feminino, enquanto que no domínio financeiro, refere-se à bolsa de valores.

Até a fase de execução dos experimentos desse trabalho, nenhuma base de dados específica de notícias financeiras foi encontrada para o treino do modelo. Portanto, os *embeddings* foram gerados por nós através de um modelo criado pelo wiki2vec[5] sobre o texto da Wikipédia em inglês, com aproximadamente 24GB. Essa ferramenta foi utilizada com o modelo *skip-gram* e com seguintes parâmetros: *size*=100, *window*=5, *min_count*=5.

Segundo (MIKOLOV et al., 2013a), o *skip-gram* é um método eficiente para aprender *embeddings* com alta qualidade capazes de capturar um grande número de relações sintáticas e semânticas entre as palavras. Os valores dos outros parâmetros dizem respeito ao tamanho dos vetores resultantes, a distância máxima entre as palavras em uma sentença e o valor mínimo de ocorrência das palavras, respectivamente (MIKOLOV et al., 2013b);

Alguns pré-processamentos também foram necessários sobre o texto da Wikipédia para que o modelo resultante fosse mais eficaz. Dentre eles, podemos enumerar:

- Remoção de pontuação;
- Remoção de caracteres especiais: Exclamação, aspas simples, aspas duplas, dois pontos, ponto e vírgula, arroba, contra barra, cerquilha, dólar, percentual, acento circunflexo, e comercial (&), asterisco, parênteses e chaves;
- Remoção dos *tokens* DBPEDIA_ID associados às entidades da Wikipédia;
- Remoção de espaços adicionais;
- Transformação de todas as palavras para caixa baixa.

5.3. CONFIGURAÇÃO

Após as etapas descritas na seção 4, cada modelo composicional gerou uma base de dados com 1080 instâncias. Essas bases de dados foram utilizadas pelos métodos de

[5] http://github.com/idio/wiki2vec

aprendizado de máquina com o objetivo de prever a polaridade do índice do dia seguinte da bolsa de valores.

A métrica utilizada para medir o desempenho das previsões foi a acurácia, também utilizada por (DING et al., 2015) cuja metodologia guiou esse trabalho. Essa métrica é calculada de acordo com a fórmula abaixo extraída de (SOKOLOVA; LAPALME, 2009), onde TP = número de verdadeiros positivos, TN = número de verdadeiros negativos, FP = número de falsos positivos e FN = número de falsos negativos.

$$Acc = \frac{TP + TN}{TP + FN + FP + TN}$$

A acurácia é uma característica de desempenho qualitativo, expressando a proximidade de concordância entre um resultado de medição e o valor do mensurando (MENDITTO; PATRIARCA; MAGNUSSON, 2007).

Os experimentos com os métodos *Naive Bayes*, KNN (K *Nearest Neighbors*), *AdaBoost*, *Bagging*, PART, J48 e *Random Forest* foram realizados utilizando o *software* Weka[6]. Para os experimentos com RNA (Rede Neural Artificial), utilizamos o *TensorFlow*[7] por possuir uma implementação eficiente e uma maior flexibilidade para configuração da arquitetura da rede. Os experimentos com SVM (*Support Vector Machine*) foram realizados utilizando o *script* easy.py[8]. Esse *script* utiliza o libSVM e faz o dimensionamento dos dados e a seleção dos parâmetros para o SVM automaticamente. Os parâmetros utilizados para cada método podem ser visualizados na Tabela 1.

[6] http://www.cs.waikato.ac.nz/ml/weka
[7] http://www.tensorflow.org
[8] http://github.com/arnaudsj/libsvm

Método	Parâmetros
Naive Bayes	batchSize=100; useKernelEstimator=False; useSupervisedDiscretization=False
KNN	K = 1, 5, 10, 12, 15; batchSize=100; crossValidate=False; windowSize=0; nearestNeighbourSearchAlgorith=LinearNNSearch
AdaBoost	batchSize=100; classifier=DecisionStump; numIterations=100; seed=1; weightThreshold=100
Bagging	bagSizePercent=100; batchSize=100; numIterations=10; seed=1
PART	batchSize=100; binarySplits=False; confidenceFactor=0.25; numFolds=3; seed=1
J48	batchSize=100; confidenceFactor=0.25; numFolds=3; seed=1
Random Forest	1) numTrees=50, 75, 100, 150, 200; maxDepth=0; numFeatures=0; seed=1 2) numTrees=100; maxDepth=5, 10; numFeatures=0; seed=1
SVM	Parâmetros escolhidos automaticamente pelo easy.py.
Rede Neural	Camadas ocultas = 2 Camada oculta 1 = 50 neurônios, *dropout* = 0.6 Camada oculta 2 = 25 neurônios Função de ativação = ReLu Algoritmo de otimização = AdamOptimizer Rodadas = 30 Épocas por rodada = 1750

TABELA 1. MÉTODOS E PARÂMETROS DE CONFIGURAÇÃO

Todos os experimentos foram executados em uma máquina com o sistema operacional Mac OS X *El Capitan* 10.11.6, processador Intel *Core* 2 *Duo* de 2.53 GHz e 4GB de RAM DDR3. Nós utilizamos validação cruzada através do método *K-Fold* com 10 partições para aumentar o poder de generalização dos métodos. Os resultados serão apresentados na seção seguinte.

5.4. RESULTADOS

O resultado de todos os experimentos executados nesse trabalho está listado na Tabela 2, onde k é o número de vizinhos mais próximos, NA é o número de árvores geradas e PM é a profundidade máxima das árvores.

Os experimentos utilizaram os métodos de aprendizado de máquina e os parâmetros informados na Tabela 1 com o objetivo de avaliar o desempenho dos modelos composicionais selecionados no domínio de previsão de preços no mercado de ações. Os resultados apresentados para a RNA estão baseados na média dos resultados obtidos em 30 rodadas, cada uma com 1750 épocas, com um intervalo de confiança de 95%, pois esse método foi o único que apresentou variação nos resultados para cada execução.

Método	Modelo composicional				
	Adição	Média	Multiplicação	*Paragraph Vector*	Concatenação
KNN (k=1)	49,44%	49,44%	53,58%	50,00%	49,07%
KNN (k=5)	50,93%	50,93%	53,02%	50,93%	49,07%
KNN (k=10)	53,52%	53,52%	54,33%	51,39%	52,69%
KNN (k=12)	53,70%	53,70%	53,77%	52,50%	52,41%
KNN (k=15)	49,91%	49,91%	51,16%	50,28%	51,39%
AdaBoost	56,02%	56,02%	55,91%	55,56%	56,57%
Bagging	52,41%	52,41%	52,09%	52,96%	53,98%
Random Forest (NA=100)	52,31%	55,00%	53,49%	53,15%	54,44%
Random Forest (NA=100, PM=5)	56,20%	56,11%	56,93%	56,94%	57,41%
Random Forest (NA=100, PM=10)	54,07%	55,28%	54,42%	55,00%	55,19%
Random Forest (NA=50)	52,78%	53,43%	55,16%	52,31%	54,63%
Random Forest (NA=75)	53,06%	53,43%	53,95%	52,22%	52,87%
Random Forest (NA= 150)	53,33%	54,07%	55,07%	54,26%	54,44%
Random Forest (NA=200)	53,61%	53,80%	54,70%	53,70%	54,72%
J48	55,09%	55,09%	55,81%	55,93%	56,20%
Naive Bayes	51,02%	51,02%	50,23%	51,20%	50,83%
libSVM	57,13%	57,13%	57,22%	57,13%	57,13%
PART	55,74%	55,74%	56,56%	57,04%	56,94%
RNA (Rede Neural Artificial)	55,87%	56,98%	**61,46%**	**64,85%**	57,98%

TABELA 2. RESULTADO DOS EXPERIMENTOS BASEADO NA ACURÁCIA

Conforme podemos notar, o modelo composicional que apresentou melhor acurácia para a base de dados utilizada foi o *Paragraph Vector*, com 64,85%, seguido do modelo composicional baseado na multiplicação com 61.46%. Podemos notar também que o desvio padrão dos resultados de cada método é baixo, exceto para a RNA, o que mostra que esse método é mais sensível às variações nas representações geradas pelos diferentes modelos composicionais.

Para calcularmos a diferença estatística entre os resultados dos modelos composicionais, aplicamos o Teste T sobre as acurácias obtidas pelos métodos de aprendizado de máquina que os utilizaram. Para o KNN e *Random Forest*, utilizamos apenas a configuração que obteve o melhor resultado, ou seja, $k = 12$ para o KNN, $NA = 100$ *e* $PM = 5$ para o *Random Forest*. O resultado pode ser visualizado na Tabela 3.

Método	Modelo composicional				
	Adição	Média	Multiplicação	*Paragraph Vector*	Concatenação
KNN	53,70%	53,70%	53,77%	52,50%	52,41%
AdaBoost	56,02%	56,02%	55,91%	55,56%	56,57%
Bagging	52,41%	52,41%	52,09%	52,96%	53,98%
Random Forest	56,20%	56,11%	56,93%	56,94%	57,41%
J48	55,09%	55,09%	55,81%	55,93%	56,20%
Naive Bayes	51,02%	51,02%	50,23%	51,20%	50,83%
libSVM	57,13%	57,13%	57,22%	57,13%	57,13%
PART	55,74%	55,74%	56,56%	57,04%	56,94%
RNA	55,87%	56,98%	**61,46%**	**64,85%**	57,98%
Média	54,80%	54,91%	55,55%	56,01%	55,50%
Valor-p	13,02%	12,38%	15,85%		27,08%

TABELA 3. TESTE T PARA OS MODELOS COMPOSICIONAIS.

Podemos notar que o *Paragraph Vector* também obteve o melhor resultado na média. No entanto, ao analisarmos os valores-p, podemos concluir que a diferença para os demais modelos composicionais não é estatisticamente significativa.

Para calcularmos a diferença estatística entre os métodos de aprendizado de máquina, seguimos o mesmo raciocínio supracitado. O resultado pode ser visualizado na Tabela 4.

Modelo composicional	Método de aprendizado de máquina								
	KNN	*AdaBoost*	*Bagging*	*Random Forest*	J48	*Naive Bayes*	libSVM	PART	RNA
Adição	53,70%	56,02%	52,41%	56,20%	55,09%	51,02%	57,13%	55,74%	55,87%
Media	53,70%	56,02%	52,41%	56,11%	55,09%	51,02%	57,13%	55,74%	56,98%
Multiplicação	53,77%	55,91%	52,09%	56,93%	55,81%	50,23%	57,22%	56,56%	61,46%
Paragraph Vector	52,50%	55,56%	52,96%	56,94%	55,93%	51,20%	57,13%	57,04%	**64,85%**
Concatenação	52,41%	56,57%	53,98%	57,41%	56,20%	50,83%	57,13%	56,94%	57,98%
Média	53,22%	56,01%	52,77%	56,72%	55,63%	50,86%	**57,15%**	56,40%	**59,43%**
Valor-p	1,32%	6,22%	0,83%	7,70%	3,37%	0,34%	11,87%	5,27%	

TABELA 4. TESTE T PARA OS MÉTODOS DE APRENDIZADO DE MÁQUINA

Podemos concluir, portanto, que para a base de dados utilizada, a RNA obteve o melhor resultado na média e quando combinada com o *Paragraph Vector*. No entanto, de acordo com os resultados do teste t, não houve diferença estatisticamente significativa em relação aos demais, uma vez que vários valores-p ficaram acima de 5%.

5.5. CONSIDERAÇÕES

De acordo com os resultados apresentados, podemos concluir os seguintes pontos a partir das hipóteses geradas no capítulo 4:

1. O *Paragraph Vector* obteve o melhor resultado na média e ao ser combinado com uma RNA. No entanto, a diferença para os demais modelos composicionais não é

estatisticamente significativa. Portanto, não podemos concluir que exista, dentre os modelos composicionais testados, o melhor modelo composicional para o domínio em questão. A conclusão a que chegamos é que deve ser escolhido o modelo composicional mais simples de ser utilizado e que exija menor poder computacional;

2. O método de aprendizado de máquina que apresentou o melhor resultado na média foi a rede neural artificial, seguido do libSVM;

3. A melhor combinação foi uma rede neural artificial, utilizando como entrada os eventos *embeddings* gerados pelo *Paragraph Vector* para representar os títulos notícias financeiras extraídas da base de dados utilizada.

Os resultados obtidos por esse trabalho corroboram com os resultados obtidos por (DING et al., 2015), nosso *baseline*. Nesse trabalho, os autores utilizaram um modelo composicional baseado em uma rede neural tensor e o modelo de geração dos *embeddings* foi treinado em uma base de notícias financeiras específica. Para prever a polaridade do índice da bolsa de valores, os autores utilizaram a influência de eventos de curto, médio e longo prazos através de uma rede neural convolutiva e obtiveram uma acurácia de 64.21%.

Conforme supracitado, nós utilizamos a mesma base de dados utilizada por (DING et al., 2015), mas treinamos o modelo de geração dos *embeddings* na Wikipédia em inglês, visto que a base de dados utilizada pelo nosso *baseline* não estava disponível. Mesmo assim, através de um modelo composicional e um método de aprendizado de máquina mais simples, obtivemos o resultado de 64.85% para a mesma tarefa de previsão.

6. CONCLUSÃO E TRABALHOS FUTUROS

A representação distribuída de palavras é uma técnica poderosa para a representação de dados que servirão de entrada para métodos de aprendizado de máquina. A partir da utilização de modelos composicionais, essa representação se torna ainda mais poderosa e abrangente, uma vez que agora podemos representar textos de tamanhos variáveis, como palavras, sentenças, parágrafos e até documentos. No entanto, ainda não existe o modelo composicional genérico que possa ser aplicado a qualquer domínio, pois, como vimos nesse trabalho, os modelos composicionais tendem a depender diretamente do domínio em que são aplicados.

Nesse trabalho, nós apresentamos uma comparação entre cinco modelos composicionais no domínio de previsão de preços no mercado de ações com o objetivo de identificar qual deles melhor representaria as notícias financeiras.

De acordo com os resultados dos experimentos, constatamos que para a base de dados utilizada, não houve diferença estatisticamente significativa entre os resultados dos modelos de composição selecionados. O *Paragraph Vector* obteve o melhor resultado, seguido do modelo baseado na multiplicação. No entanto, ao observarmos os métodos de aprendizado de máquina escolhidos, identificamos que a rede neural apresentou os melhores resultados tanto na média quanto ao utilizar as representações geradas pelo *Paragraph Vector*. Esses resultados mostram que a combinação entre rede neural artificial e eventos *embeddings* é poderosa e promissora.

Como trabalhos futuros, nós pretendemos conduzir pesquisas adicionais em modelos composicionais, comparando aqueles mais complexos baseados em aprendizado como LSTM (*Long Short-Term Memory*), *Auto Encoder*, Rede Neural Tensor e etc. no mesmo domínio desse trabalho com o objetivo de tentar melhorar ainda mais os resultados da previsão e, possivelmente, desenvolver um novo modelo composicional. Além disso, pretendemos investigar o impacto do reconhecimento prévio de entidades nomeadas na base de dados de notícias financeiras na representação desses modelos composicionais.

7. BIBLIOGRAFIA

AGRAWAL, J. G.; CHOURASIA, V. S.; MITTRA, A K. State-of-the-Art in Stock Prediction Techniques. Advanced Research in Electrical and Instrumental Engineering, v. 2, n. 4, p. 1360–1366, 2013.

ATSALAKIS, G. S.; VALAVANIS, K. P. Surveying stock market forecasting techniques - Part II: Soft computing methods. Expert Systems with Applications, v. 36, n. 3 PART 2, p. 5932–5941, 2009.

BANKO, M. et al. Open Information Extraction from the Web. Proceedings of IJCAI-07, the International Joint Conference on Artificial Intelligence, p. 2670–2676, 2007a.

BANKO, M. et al. TextRunner: Open Information Extraction on the Web. Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT), p. 25–26, 2007b.

BLACOE, W.; LAPATA, M. A Comparison of Vector-based Representations for Semantic Composition. Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL '12), n. July, p. 546–556, 2012.

CHEN, D. et al. Neural Tensor Networks and Semantic Word Vectors. International Conference on Learning Representations, p. 1–4, 2013.

DEERWESTER, S. et al. Indexing by latent semantic analysis. Journal of the American Society for Information Science, v. 41, n. 6, p. 391–407, 1990.

DING, X. et al. Using Structured Events to Predict Stock Price Movement: An Empirical Investigation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, p. 1415–1425, 2014.

DING, X. et al. Deep Learning for Event-Driven Stock Prediction. International Joint Conference on Artificial Intelligence, n. Ijcai, p. 2327–2333, 2015.

DJURIC, N. et al. Hate Speech Detection with Comment Embeddings. Proceedings of the 24th International Conference on World Wide Web - WWW '15 Companion, v. 32, p. 29–30, 2015.

ELMING, JAKOB AND JOHANNSEN, ANDERS AND KLERKE, SIGRID AND LAPPONI, EMANUELE AND ALONSO, HECTOR MARTINEZ AND SOGAARD, A. Down-stream effects of tree-to-dependency conversions. Hlt-Naacl, n. June, p. 617–626, 2013.

ETZIONI, O. et al. Open Information Extraction: The Second Generation. Ijcai, v. 11, p. 3–10, 2011.

FADER, A.; SODERLAND, S.; ETZIONI, O. Identifying relations for open information extraction. Proceedings of the Conference on …, p. 1535–1545, 2011.

FAN, M.; ZHOU, Q.; ZHENG, T. F. Learning Embedding Representations for Knowledge Inference on Imperfect and Incomplete Repositories. Proceedings - 2016 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2016, p. 42–48, 2017.

FRANK, E.; WITTEN, I. H. Generating accurate rule sets without global optimization. Proceedings of the Fifteenth International Conference on Machine Learning, p. 144–151, 1998.

HARRINGTON, P. Machine learning in action. 2012.

HASSAN, M. R.; NATH, B. Stock market forecasting using hidden Markov model: a new approach. Intelligent Systems Design and Applications, 2005. ISDA '05. Proceedings. 5th International Conference on, p. 192–196, 2005.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. The Elements of Statistical Learning. Elements of Statistical Learning, 2009.
HERMANN, K. M.; BLUNSOM, P. Multilingual Models for Compositional Distributed Semantics. Acl, p. 58–68, 2014.
IYYER, M. et al. Deep Unordered Composition Rivals Syntactic Methods for Text Classification. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), v. 53, n. 9, p. 1689–1699, 2015.
JAIN, A. K.; MAO, J.; MOHIUDDIN, K. M. Artificial neural networks: A tutorial. Computer, v. 29, n. 3, p. 31–44, 1996.
KALCHBRENNER, N.; GREFENSTETTE, E.; BLUNSOM, P. A Convolutional Neural Network for Modelling Sentences. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, p. 655–665, 2014.
LANDAUER, T. K.; DUMAIS, S. T. A solution to Plate's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Physcol. Rev., v. 104, n. 2, p. 211–240, 1997.
LE, Q.; MIKOLOV, T. Distributed representations of sentences and documents. Proceedings of the 31st International Conference on, 2014.
LEV, G.; KLEIN, B.; WOLF, L. In defense of word embedding for generic text representation. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), v. 9103, p. 35–50, 2015.
LIDDY, E. D. Natural Language Processing. Natural Language Processing, 2001.
LIU, Y. et al. Topical Word Embeddings. Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15), v. 2, n. C, p. 2418–2424, 2015.
LUSS, R.; D'ASPREMONT, A. Predicting abnormal returns from news using text classification. Quantitative Finance, v. 15, n. 6, p. 999–1012, 2009.
MAUSAM et al. Open Language Learning for Information Extraction. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Jeju Island, Korea, 12–14 July 2012, n. July, p. 523–534, 2012.
MENDITTO, A.; PATRIARCA, M.; MAGNUSSON, B. Understanding the meaning of accuracy, trueness and precision. Accreditation and Quality Assurance, v. 12, n. 1, p. 45–47, 2007.
MICHALSKI, R. S.; CARBONELL, J. G.; MITCHELL, T. M. Machine Learning: An Artificial Intelligence approach. p. 572, 2013.
MIKOLOV, T. et al. 10.1162/Jmlr.2003.3.4-5.951. CrossRef Listing of Deleted DOIs, v. 1, p. 1–9, 2000.
MIKOLOV, T. et al. Distributed representations of words and phrases and their compositionality. Advances in neural, 2013a.
MIKOLOV, T. et al. Efficient Estimation of Word Representations in Vector Space. Proceedings of the International Conference on Learning Representations (ICLR 2013), p. 1–12, 2013b.
MIKOLOV, T.; YIH, W.; ZWEIG, G. Linguistic regularities in continuous space word representations. hlt-Naacl, 2013.
MITCHELL, J.; LAPATA, M. Vector-based Models of Semantic Composition. ACL, 2008.
MITCHELL, J.; LAPATA, M. Composition in distributional models of semantics. Cognitive science, v. 34, n. 8, p. 1388–1429, 2010.
MITTERMAYER, M. A.; KNOLMAYER, G. F. NewsCATS: A news categorization and trading system. Proceedings - IEEE International Conference on Data Mining, ICDM, p. 1002–1007, 2006.

PALANGI, H. et al. Deep Sentence Embedding Using Long Short-Term Memory Networks: Analysis and Application to Information Retrieval. arXiv:1502.06922 [cs], v. 24, n. 4, p. 1–25, 2015.
RADINSKY, K.; DAVIDOVICH, S.; MARKOVITCH, S. Learning causality for news events prediction. Proceedings of the 21st international conference on World Wide Web (WWW2012), n. 909–918, p. 909, 2012.
RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. Nature, v. 323, n. 6088, p. 533–536, 1986.
SOCHER, R.; HUANG, E.; PENNINGTON, J. Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection. Advances in Neural Information Processing Systems, p. 801–809, 2011.
SOKOLOVA, M.; LAPALME, G. A systematic analysis of performance measures for classification tasks. Information Processing and Management, v. 45, n. 4, p. 427–437, 2009.
TAI, K. S.; SOCHER, R.; MANNING, C. D. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. Proceedings of ACL, p. 1556–1566, 2015.
TETLOCK, P. C. Giving content to investor sentiment: The role of media in the stock market. Journal of Finance, v. 62, n. 3, p. 1139–1168, 2007.
WELD, D. S.; WU, F. Open Information Extraction using Wikipedia. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, n. July, p. 118–127, 2010.
WIETING, J. et al. Towards Universal Paraphrastic Sentence Embeddings. Under review of ICLR, p. 1–17, 2016.
WITTEN IAN H AND FRANK, EIBE AND HALL, MARK A AND PAL, C. J. Data Mining: Practical Machine Learning Tools and Techniques. Chapter 7. p. 1–45, 2016.
XIE, B.; PASSONNEAU, R. J.; WU, L. Semantic Frames to Predict Stock Price Movement. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, p. 873–883, 2013.
YANG, B. et al. Learning Multi-Relational Semantics Using Neural-Embedding Models. Nips, p. 1–5, 2014.
YU, M.; DREDZE, M. Learning Composition Models for Phrase Embeddings. Transactions of the ACL, v. 3, p. 227–242, 2015.
ZHU, J. et al. StatSnowball : a Statistical Approach to Extracting Entity. Proceedings of the 18th international conference on World wide web (WWW2009), p. 101–110, 2009.