Pós-Graduação em Ciência da Computação

Thiago Fernandes de Sousa

Combinação de classificadores para sistema de *Automated Fact Checking*

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
https://www3.cin.ufpe.br/br/pos-graduacao/mestrado-profissional/sobre-o-mestrado-profissional

Recife
2020

Thiago Fernandes de Sousa

Combinação de classificadores para sistema de *Automated Fact Checking*

Dissertação de mestrado apresentada ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Área de Concentração: Inteligência computacional
Orientador: George Darmiton da Cunha Cavalcanti

Recife
2020

Catalogação na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

S725c Sousa, Thiago Fernandes de
Combinação de classificadores para sistema de *automated fact checking* / Thiago Fernandes de Sousa. – 2020.
87 f.: il., fig., tab.

Orientador: George Darmiton da Cunha Cavalcanti.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2020.
Inclui referências e apêndices.

1. Inteligência computacional. 2. Combinação de classificadores. I. Cavalcanti, George Darmiton da Cunha (orientador). II. Título.

006.31 CDD (23. ed.) UFPE - CCEN 2020 - 133

Thiago Fernandes de Sousa

Combinação de classificadores para sistema de *Automated Fact Checking*

Dissertação apresentada ao Programa de Pós-Graduação Profissional em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre Profissional em 26 de maio de 2020.

Aprovado em: 26 / 05 / 2020.

BANCA EXAMINADORA

_____

Profa. Patricia Cabral de Azevedo Restelli Tedesco
Centro de Informática / UFPE

_____

Prof. Rinaldo José de Lima
Universidade Federal Rural de Pernambuco

_____

Prof. George Darmiton da Cunha Cavalcanti
Centro de Informática / UFPE
(Orientador)

Dedico este trabalho a todos que entendem a ciência como a melhor ferramenta conhecida de busca da verdade.

AGRADECIMENTOS

A todos meus professores.

RESUMO

A propagação de notícias falsas se tornou um problema de proporções globais, afetando a economia, saúde pública, convívio social, relações internacionais e o processo eleitoral de diversos países. Estudos indicam que *fake news* são compartilhadas mais vezes, e de maneira mais rápida que notícias verdadeiras. Isso ocorre porque estas declarações são fabricadas para enganar o leitor, indo de encontro com as suas convicções pessoais e diminuindo o seu senso crítico. Diversas técnicas de *machine learning* vêm sendo empregadas na tentativa de identificar padrões existentes em *fake news*, criando assim os sistemas de *Automated Fact Checking*. Uma alternativa a se considerar na melhoria de qualquer problema de classificação, é a combinação de um grupo de classificadores para uma classificação em conjunto, abrindo a possibilidade de se combinar os acertos individuais de cada integrante do grupo, obtendo assim, um resultado na classificação em conjunto que supere os resultados individuais de cada membro do conjunto. No entanto, combinar um grupo de classificadores, de forma a conseguir com que estas técnicas se complementem, não é uma tarefa trivial. Tendo em vista que para se conseguir uma melhoria no desempenho, os classificadores participantes do conjunto devem apresentar variações no seus padrões de acertos e erros. Este trabalho propõe uma abordagem que, dado um *pool* de classificadores, seja possível analisar o comportamento de cada integrante do conjunto em relação a todos os outros, tornando viável a construção de subgrupos de classificadores que apresente uma boa diversidade entre seus membros. Para testar a abordagem proposta, foi construído um *pool* composto por 80 classificadores, que tiveram seus desempenhos individuais verificados na classificação de um conjunto de dados de *fake news*. Em seguida, foi aplicado a metodologia proposta, e selecionados para uma classificação conjunta, subgrupos que apresentaram melhor diversidade entre seus membros. Este processo foi realizado duas vezes, a primeira vez considerando uma classificação binária do problema, e na segunda, foram consideradas seis classes diferentes, cada uma relacionada ao nível de veracidade contido na declaração analisada. Em todos os casos analisados, a aplicação da proposta se mostrou eficiente, possibilitando encontrar subgrupos que apresentaram melhora de desempenho na classificação em conjunto quando comparados com o desempenho individual dos classificadores do *pool*, superando também, experimentos publicados em outros trabalhos que se dedicavam a classificar o mesmo conjunto de dados.

Palavras-chaves: Combinação De Classificadores. *Automated Fact-Checking*. Detecção De *Fake News*.

ABSTRACT

The spread of false news has become a problem of global proportions, affecting the economy, public health service, social life, international relations and the electoral process of several countries. Studies indicate that fake news is shared more often, and faster than real news. This is because these statements are designed to deceive the reader, meeting their personal beliefs and diminishing their critical sense. Several machine learning techniques have been used in an attempt to identify existing patterns in fake news, thus creating Automated Fact Checking systems. An alternative to consider in the improvement of any classification problem, is the combination of a group of classifiers for a classification together, opening the possibility of combining the individual hits of each member of the group, thus obtaining a result in the classification in set that exceeds the individual results of each member of the set. However, combining a group of classifiers in order to make these techniques complement each other is not a trivial task. Bearing in mind that to achieve an improvement in performance, the classifiers participating in the set must present variations in their patterns of successes and errors. This work proposes an approach that, given a pool of classifiers, it is possible to analyze the behavior of each member of the set in relation to all the others, making it possible to construct subgroups of classifiers that present a good diversity among its members. To test the proposed approach, a pool composed of 80 classifiers was built, which had their individual performances verified in the classification of a fake news data set. Then, the proposed methodology was applied, and subgroups that showed the best diversity among their members were selected for a joint classification. This process was carried out twice, the first time considering a binary classification of the problem, and the second time, six different classes were considered, each related to the level of veracity contained in the analyzed statement. In all the cases analyzed, the application of the proposal proved to be efficient, making it possible to find subgroups that showed improved performance in the classification as a whole when compared to the individual performance of the classifiers in the pool, also surpassing experiments published in other works that were dedicated to classifying the same data set

Keywords: Ensemble Of Classifiers. Automated Fact-Checking. Fake News Detection.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ADB	AdaBoost
AFC	*Automated Fact-Checking*
BNB	BernoulliNB
BoW	*Bag-of-Words*
CNB	ComplementNB
CNN	Convolutional Neural Networks
CV	CountVectorizer
ET	ExtraTrees
FT	FastText
GL	GloVe
GNB	GaussianNB
KNN	K Nearest Neighbor
LR	LogisticRegression
LSTM	Long Short Term Memor
MCS	*Multiple Classifier Systems*
MDS	*Multidimensional Scaling*
MLP	Multi-layer Perceptron
MNB	MultinomialNB
NLP	*Natural Language Processing*
P	Perceptron
PA	Passive Aggressive
PCA	*Principal Component Analysis*
RF	RandomForest
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine - Linear SVC
t-SNE	*t-Distributed Stochastic Neighbor Embedding*
TFIDF	*Term Frequency Inverse Document Frequency*
UMAP	*Uniform Manifold Approximation and Projection*
W2V	Word2Vec

LISTA DE SÍMBOLOS

Γ	Conjunto de dados de treinamento
Θ	Conjunto de dados de validação

SUMÁRIO

1 INTRODUÇÃO

A popularização da internet, associada ao barateamento de tecnologias e a crescente facilidade do compartilhamento de informações, causou uma revolução social sem precedentes na história. O aumento no número de canais de notícias não tradicionais, fruto desta revolução, fez surgir um número alarmante de notícias falsas compartilhadas, causando forte impacto negativo na economia, política, saúde pública e no convívio social(SHABAN; HEXTER; CHOI, 2017; LAZER et al., 2018; CINELLI et al., 2020).

Estudos apontam que *fake news* foram massivamente usadas na crise entre Ucrânia e Rússia ocorrida em 2015, tiveram também influência na corrida presidencial americana de 2016, bem como no "Brexit", referendo que discutia a saída do Reino Unido da União Europeia, ocorrido naquele mesmo ano (ZANNETTOU et al., 2019; ROSE, 2017; VOLKOVA et al., 2017; KHALDAROVA; PANTTI, 2016). Foram observadas também um grande número de notícias falsas sobre a pandemia global de COVID-19 circulando nas redes sociais (CINELLI et al., 2020), considerando o volume de informações disseminadas, a gravidade, e o potencial destrutivo de *fake news* relacionados a este assunto, pesquisadores já classificam que o combate a notícias falsas sobre o Sars-CoV-2 é um dos desafios impostos à humanidade pelo vírus, e a comunidade de *machine learning* já começa a apresentar estudos e conjuntos de dados dedicados a este tema. (LAI et al., 2020; CUI; LEE, 2020). A divulgação de *fake news* modificou drasticamente as estratégias das campanhas eleitorais, recebendo grande destaque nos meios de comunicação (JR.; LIM; LING, 2018; MORGAN, 2018; SHAO et al., 2017; PERSILY, 2017). Há fortes indicativos de participação de empresas privadas nestes processos, aplicando técnicas de *machine learning* para ampliar e otimizar o uso de *fake news*, identificando perfis de usuários mais suscetíveis a determinados temas e usando *boots*, de forma a direcionar a opinião pública de acordo com os interesses do contratante, isolando o receptor em uma bolha de opiniões, onde suas convicções não são confrontadas (SPOHR, 2017; Isaak; Hanna, 2018; ARNAUDO, 2017; ALLCOTT; GENTZKOW, 2017). Uma evidência indireta da relevância deste tema é o fato de que a maioria dos jornais de grande circulação no mundo contam hoje com uma coluna dedicada a checagem de notícias publicadas em redes sociais ou em mídias não tradicionais. Todos estes indicativos demonstram que a disseminação de notícias falsas tem se tornado um problema global capaz de influenciar decisões políticas importantes, afetando relações internacionais entre potências nucleares e causando problemas sociais graves.

Há uma dificuldade implícita em humanos detectarem notícias falsas, tendo em vista que as mesmas são construídas com o objetivo de confundir o receptor, corroborando com suas convicções pessoais, diminuindo assim a imparcialidade no julgamento de uma determinada informação (LAZER et al., 2018). Outro agravante a este cenário é a velocidade que uma *fake news* pode se disseminar. Estudos feitos na rede social Twitter apontaram que

notícias falsas são compartilhadas mais rapidamente, e por mais vezes que notícias verdadeiras, em especial quando estas notícias estão associadas a fatos políticos envolvendo opinião e interpretação de fatos (VOSOUGHI; ROY; ARAL, 2018).

Uma das formas encontradas de combater a propagação de *fake news* é o uso de inteligência artificial com o objetivo de identificar padrões presentes em afirmações falsas e realizar um processo de classificação, enquadrando a declaração em uma classe relacionada ao nível de veracidade presente no texto. Estes sistemas são conhecidos como *Automated Fact Checking* e vem sendo tema de grande notoriedade em publicações científicas (CONROY; RUBIN; CHEN, 2015; WANG, 2017; SPOHR, 2017). Atualmente a comunidade científica que se dedica a este tema vem testando diversas técnicas, e não há um consenso sobre a melhor abordagem para o trato deste problema. Considerando que existe um grande número de opções disponíveis, e que cada proposta traz consigo vantagens e limitações, o uso combinado de mais de uma técnica de classificação, buscando que estas técnicas se completem, é uma opção que se torna interessante. Um sistema de classificação combinado, que usa mais de uma técnica de classificação ao mesmo tempo, é composto por um grupo de classificadores, denominado *pool*, e uma regra que define a interação entre os membros do *pool*. Para se obter sucesso no uso desta técnica, é determinante construir um *pool* adequado, em que cada membro contribua para o desempenho final da combinação. Uma falha na escolha dos classificadores participantes do *pool* compromete todo o processo, tornando impossível obter uma melhora no desempenho da classificação, em outras palavras, errar na composição do *pool* torna toda técnica inócua, independente da regra de combinação selecionada. Considerando a enorme variedade de classificadores, suas variações de configuração, e ainda, a mudança de comportamento de um mesmo classificador quando este é utilizado em conjunto com extratores de características diferentes, em problemas de classificação que envolvam processamento de linguagem natural, montar um *pool* adequado é uma tarefa complicada, que consome tempo, e diversos testes são necessários para se encontrar a composição ideal. Ter uma ferramenta que apresenta de maneira visual a interação entre os classificadores de um grupo pode facilitar este trabalho e economizar recursos humanos e computacionais.

Este trabalho inicia suas contribuições para com a comunidade científica observando que, partindo de um *baseline* apresentado em outros trabalhos, há possibilidade de melhorar o *state of art* do problema de detecção automática de *fake news* por meio da combinação de classificadores. Entendendo ainda que a seleção do grupo de classificadores é uma tarefa delicada e de difícil execução, e ao mesmo tempo decisiva na implementação de uma classificação combinada, é apresentada uma abordagem que facilita a seleção de grupos de classificadores mais adequados para uma classificação conjunta, entregando uma maneira visual de identificar o comportamento de cada classificador candidato a participar do *pool*, abrindo a possibilidade de eliminar membros que não contribuem na melhoria de desempenho do conjunto, e apresentando como saída os subgrupos mais promissores para uma

classificação conjunta. São aplicadas também técnicas de combinação nos conjuntos de classificadores selecionados, os resultados obtidos no final deste processo são apresentados à comunidade como uma contribuição, ao mesmo tempo em que validam a proposta do trabalho, que é a contribuição principal desta pesquisa.

1.1 MOTIVAÇÃO

Combater a propagação de notícias falsas não é uma tarefa trivial, e diversas soluções vem sendo propostas por jornalistas, economistas e comunidade científica (BAKIR; MCSTAY, 2018; SULLIVAN, 2019). Existem algumas diferenças entre o problema de detecção de *fake news* e os demais problemas de classificação de texto. Uma particularidade que fica evidente ao analisar a literatura é a complexidade. Em problemas envolvendo classificação de texto, é comum que os modelos apresentarem uma acurácia de mais de 80%, no entanto, para problemas como o abordado nesta dissertação, o *baseline* fica pouco acima do que se espera de uma classificação aleatória (Sarakit et al., 2015; WANG, 2017; HORNE; ADALI, 2017). Considerando que até mesmo a definição formal de *fake news* ainda é tema de discussão, montar um conjunto de dados não enviesado é outra dificuldade relacionada a este tema, e diversas estratégias diferentes são adotadas para tentar contornar este problema (SALEM et al., 2019; WANG, 2017). Mesmo com as dificuldades apresentadas, a eficiência do uso de inteligência artificial em diversos outros problemas de reconhecimento de padrões é um indicativo de que a aplicação destas técnicas pode contribuir com a solução do problema e abordagens que fazem o uso de *machine learning* são cada vez mais sugeridas (KUCHARSKI, 2016; VLACHOS, 2018; GRAVES, 2018).

Os sistemas de *Automated Fact-Checking* (AFC) aplicam técnicas de *machine learning* para, recebendo uma declaração como valor de entrada, identificar as principais características presentes no texto e classificar esta entrada de acordo com seu nível de veracidade (GRAVES, 2018; HASSAN et al., 2015). Os dois principais componentes de um sistema de AFC são: a técnica de extração de características, responsável por transformar o texto em uma representação numérica, e o algoritmo classificador, responsável por ler esta representação numérica e a classificar em um número limitado de classes, indicando se o texto é verdadeiro, falso, ou se encontra em algum ponto intermediários entre estas classes. As técnicas de extração de características são objeto de estudo da comunidade de processamento de linguagem natural, e tem por objetivo fornecer aos computadores representações compreensíveis de textos ou falas proferidas por humanos, existem diversas técnicas disponíveis para esta tarefa, indo desde as representações mais simples, baseadas em modelos *Bag of Words*, onde o texto é representado por uma coleção de palavras não ordenadas, até as modernas redes neurais profundas, que levam em conta o contexto semântico de cada palavra dentro da oração. Já os classificadores, são algoritmos que fazem uso de modelos matemáticos para criar associações entre as características, disponibilizadas pelos extratores, e uma classe.

Dentro de um sistema de AFC os extratores de características têm forte influência no comportamento do algoritmo classificador, sendo que a escolha de uma técnica adequada é de fundamental importância. Diversos extratores de caraterísticas e algoritmos classificadores tem sido avaliadas pela comunidade científica, sendo que uma opção a ser considerada na melhora do *state of the art* de qualquer problema de classificação, é o uso combinado de mais de uma técnica de classificação.

Um sistema de classificação em conjunto é composto de um grupo de classificadores, e uma função que define a regra de combinação entre estes classificadores, sendo que cada membro do grupo recebe como entrada um padrão, e indica a qual classe aquele padrão pertence e a regra de combinação vai avaliar a resposta de cada classificador e definir qual a decisão do conjunto, indicando a classificação final do padrão. O uso combinado de técnicas distintas de classificação não é uma garantia na melhoria de resultados, para o sucesso da abordagem é importante que os classificadores envolvidos no processo se complementem, apresentando padrões de acertos e erros distintos entre si. Um conjunto em que todos os classificadores se comportem de maneira igual vai resultar em uma classificação conjunta idêntica a classificação individual de cada membro, independente da função que combine os resultados destes classificadores, tornando portanto, esta combinação ineficiente (KUNCHEVA; WHITAKER, 2003). Considerando que haja diversidade na resposta dos classificadores do *pool*, é importante também selecionar uma regra de combinação adequada, que consiga agregar os acertos individuais de cada classificador do grupo. Estes dois fatores, *pool* e regra de combinação, são determinantes para o sucesso de uma classificação combinada. Diversos trabalhos demonstram que a implementação adequada de uma combinação de classificadores podem apresentar melhoria nos resultados se comparado com abordagens monolíticas em problemas variados (Kittler et al., 1998). Existem publicações dedicadas exclusivamente ao estudo de combinação de técnicas de classificação para o problema de detecção de *fake news* (THORNE et al., 2017), e não é incomum em publicações que fazem o uso de outras abordagems testarem algum tipo de combinação entre os classificadores analisados para fins de comparação (WANG, 2017; ALHINDI; PETRIDIS; MURESAN, 2018).

Observando os resultados apontados pela literatura, que indicam que AFC podem contribuir para a solução do problema de detecção de *fake news*, bem como o sucesso de técnicas de combinação de classificadores em outros problemas, é intuitivo estudar maneiras mais adequadas de explorar tais técnicas em um problema de relevância tão evidente como o apresentado.

1.2 OBJETIVO

Este trabalho se dedica a explorar o problema de detecção automática de *fake news* usando combinações de técnicas distintas de classificação, propondo uma maneira visual de

encontrar os subgrupos mais promissores dentro de um *pool* de classificadores, e analisando as técnicas de combinação dentro destes subgrupos.

1.3 OBJETIVOS ESPECÍFICOS

- Analisar a viabilidade do uso de combinação de classificadores no problema de detecção de *fake news*;
- Desenvolver uma técnica visual de seleção de grupos de classificadores mais promissores para uma classificação conjunta;
- Executar experimentos para validar os métodos propostos e avaliar seus desempenhos frente aos métodos do *state of the art*;

1.4 APRESENTAÇÃO

Este trabalho está dividido em cinco partes, o Capítulo 1 apresenta o problema e as motivações que levaram a esta pesquisa. No Capítulo 2 é feito um levantamento do estado atual do estudo do problema na comunidade científica, e são apresentados alguns projetos de destaque relacionados ao tema. O Capítulo 3 discute alguns conceitos básicos fundamentais para o entendimento deste trabalho. No Capítulo 4 é apresentada uma proposta de experimento, que posteriormente é testada e tem seus resultados apresentados no Capítulo 5. O Capítulo 6 apresenta as conclusões desta pesquisa e discute algumas sugestões de trabalhos futuros.

2 TRABALHOS RELACIONADOS

Demonstrada a relevância do problema abordado na proposta desta dissertação, é fácil observar que o assunto vem ganhando espaço na discussão acadêmica, e diversas iniciativas dedicadas a entender, documentar e propor soluções para o problema da disseminação de *fake news* em mídias digitais vem sendo propostas pela comunidade científica. A Seções 2.1 apresentam as principais iniciativas e projetos relacionados ao tema, e na Seção 2.2 é feita uma análise do estado da arte do problema abordado.

2.1 PROJETOS E PESQUISAS

No ano de 2017 um consórcio jornalístico propôs um desafio para a comunidade científica que se dedicava ao estudo de técnicas de detecção automática de *fake news*, oferecendo prêmios às equipes que apresentassem as melhores abordagens de classificação para um conjunto de dados proposto. A competição ganhou o nome de *Fake News Challenge*[1] e contou com grande adesão da comunidade de *machine learning*, sendo posteriormente pauta de diversas publicações que se dedicaram a fazer análise das estratégias adotadas pelas equipes mais bem sucedidas. Em geral, houve uma grande variação na escolha dos algoritmos classificadores e extratores de características, sendo experimentadas diversas abordagens, no entanto, a equipe *SOLAT in the SWEN* conseguiu o melhor desempenho entre todos os participantes fazendo uso combinado de dois classificadores, que por sua vez, tiveram o seus funcionamentos atrelados a diferentes extratores de características. Este bom resultado indica o potencial de técnicas de combinação dentro de um problema de classificação reconhecidamente complexo (RIEDEL et al., 2017; HANSELOWSKI et al., 2018). Outros três projetos podem ser destacados para demonstrar o empenho acadêmico no trato do problema. A iniciativa brasileira Eleições Sem Fake[2], encabeçada pelo departamento de informática da UFMG em parceria com a USP, se dedica a monitorar e traçar padrões de espalhamento de notícias falsas em mais de 500 grupos públicos de whatsapp brasileiros. Publicações resultantes da análise cuidadosa de troca de mensagens entre participantes destes grupos revelam o forte viés político dos mesmos, características presentes em quase todas os grupos analisados (RESENDE et al., 2018; RIBEIRO, 2018). Um outro projeto que se dedica ao mesmo tema, é o FANDANGO[3], iniciativa que tem como objetivo melhorar a qualidade das informações disponíveis para o cidadão europeu, servindo de suporte científico para a indústria jornalística daquele bloco. Como resultado, esta iniciativa vem apresentando grande número de estudos, e reunindo dados relativos a propagação de *fake news* na União Europeia, servindo de fonte de pesquisas para profissionais de diferentes

[1] http://www.fakenewschallenge.org/
[2] https://www.eleicoes-sem-fake.dcc.ufmg.br/
[3] https://fandango-project.eu

ramos do conhecimento que se dedicam ao entendimento e combate do problema. Por fim, destaca-se também o projeto FEVER[4], Workshop anual, organizado e patrocinado pela Amazon em parceria com a Universidade de Cambridge, que apresenta um conjunto de dados em forma de desafio para pesquisadores da área de *machine learning*, e tem contribuído com a comunidade de pesquisa sendo citado em várias publicações (THORNE et al., 2018).

2.2 ESTADO DA ARTE DO PROBLEMA

Como demonstrado na Seção 2.1, diversas frentes vem estudando e testando estratégias distintas, e ainda não existe consenso na comunidade científica de qual a melhor abordagem para a classificação de *fake news*. Uma maneira de observar o estado da arte do problema é, selecionar um conjunto de dados relevante para a comunidade, e analisar a evolução das estratégias adotadas para classificá-lo por meio de uma linha do tempo. A Tabela 1 sintetiza uma análise do Liar Dataset, apresentando as principais publicações relacionadas a ele, bem como uma descrição da estratégia adotada em cada trabalho. Este conjunto de dados foi apresentado por Wang (WANG, 2017), que já na sua publicação inicial, estabeleceu um *baseline* analisando o comportamento de alguns classificadores, o melhor resultado foi obtido usando a rede neural Convolutional Neural Networks (CNN), que recebendo somente o texto principal da notícia obteve um desempenho de 27,0% e recebendo todos os campos disponíveis no conjunto de dados apresentou desempenho de 27,4%. Ainda no ano de 2017, Bhattacharjee, Talukder e Balantrapu (Das Bhattacharjee; Talukder; Balantrapu, 2017) apresentaram experimentos usando o mesmo conjunto de dados colocando o foco no uso das redes neurais CNN e Long Short Term Memor (LSTM), no entanto, os resultados apresentados neste trabalho não são passíveis de comparação, pois, a estratégia adotada para treinamento envolvia o uso de um segundo conjunto de dados. Estratégia semelhante à adotada por Kirilin e Strube em 2018, que conseguiram bons resultados cruzando as declarações contidas no Liar com um conjunto de dados de credibilidade dos autores de cada declaração (KIRILIN; STRUBE, 2018). No mesmo ano Alhindi (ALHINDI; PETRIDIS; MURESAN, 2018) apresentou um trabalho no qual realizava experimentos abordando o conjunto de dados de forma binária e com múltiplas classes, apresentando os resultados de 61,0% para o problema binário e 23,0% para o problema com seis classes, recebendo como entrada somente o texto principal da notícia, já o melhor resultado considerando como entrada o texto principal da notícia em conjunto com outros metadados foi de e 70,0% para o problema binário e 36,0% para o problema com múltiplas classes.

[4] https://fever.ai/

Jul 2017	Apresentação do Liar Dataset, melhor resultado 27,7% (WANG, 2017).
Set 2017	Testes com redes neurais CNN e LSTM, uso do Harverd Dataset para treinamento dos classificadores(Das Bhattacharjee; Talukder; Balantrapu, 2017).
Mar 2018	Cruzamento de informações entre os datasets Liar e Speaker2credit(KIRILIN; STRUBE, 2018).
Ago 2018	Experimentos com rede neural MMFD (KARIMI et al., 2018).
Nov 2018	Análise binária do dataset, adição de metadados(ALHINDI; PETRIDIS; MURESAN, 2018).
Ago 2019	Experimentos com análise de sentimentos no Liar Dataset(Bhutani et al., 2019).
Jan 2020	Experimentos redes neurais diversas (DEBNATH et al., 2020).

Tabela 1 – Linha do tempo com principais publicações que realizaram experimentos com o Liar Dataset.

Em geral, para melhorar o desempenho na classificação, os pesquisadores colocaram ênfase em testes com redes neurais diversas, e adição de valores extras no treinamento dos classificadores, ainda assim, analisando os melhores resultados apresentados, fica evidente a complexidade da tarefa. Uma segunda conclusão proveniente da análise dos trabalhos publicados, é que ainda não há um consenso da melhor abordagem para o problema, o que justifica a experimentação de novas estratégias.

2.3 ADIÇÃO DE FEATURES

Trabalhos diferentes adotaram estratégias diversas para melhorar o desempenho dos classificadores, sendo que cada abordagem traz consigo vantagens e limitaçõs provenientes de sua natureza. No entanto, algo bastante recorrente em publicações que se dedicam a realizar experimentos tratando de problemas semelhantes ao abordado neste trabalho, é o uso de *features* adicionais, características não presentes nos padrões originais. A análise de sentimentos, por exemplo, é uma estratégia que tem se mostrado bastante eficiente, trata-se da aplicação de algorítmos que, recebendo como entrada um texto, retornar um valor numérico associado à presença e intensidede de determinadas emoções humanas na

informação analisada. Em seguida, estes valores são adicionados as *features* já presentes no conjunto de dados, e utilizados no processo de treinamento dos classificadores. Como discutido no Capítulo 1, é comum que *fake news* tragam consigo uma grande carga de emoções, o que torna a análise de sentimento uma técnica particularmente adequada para este problema (Reis et al., 2019; GUO et al., 2019; Bhutani et al., 2019).

Outra estratégia que tem se mostrado interessante é o uso de abordagens linguísticas e baseadas em rede, técnicas que apresentam alta precisão na classificação em domínios limitados. Comumente aplicadas a conjunto de dados compostos de informações provenientes de redes sociais, analisando o padrão de compartilhamento de uma notícia dentro do FaceBook ou Twitter, por exemplo, é possível identificar características comuns a notícias falsas e verdadeiras, e quando esta *features* é disponibilizada para um classificador, pode influenciar na melhora do seu desempenho (CONROY; RUBIN; CHEN, 2015).

3 CONCEITOS BÁSICOS

Este Capítulo apresenta os principais conceitos para o entendimento da proposta desta dissertação. A Seção 3.1 Apresenta os conceitos básicos de aprendizado de máquina, a Seção 3.2 trata das técnicas de representação de texto utilizadas nos experimentos, a Seção 3.3 discute sobre técnicas de *Ensemble* que fazem uso de *pool* homogênio, e suas aplicações neste trabalho, a Seção 3.4 discute o funcionamento de um sistema de combinação de classificadores e nas Seções 3.5 e 3.6 são discutidos respectivamente o conceito de Oracle e técnicas de redução de dimensionalidade.

3.1 APRENDIZADO DE MÁQUINA SUPERVISIONADO

Aprendizado de máquina é a subárea da inteligência artificial que se dedica ao estudo de técnicas que possibilitam aos computadores realizarem tarefas para as quais eles não foram explicitamente programados. Uma segmentação desta área é o de aprendizado de máquina supervisionado, conjunto de técnicas que exige um processo prévio de treinamento dos algoritmos, e só então, os mesmos estão prontos para realizar a tarefa para as quais eles foram desenvolvidos. O processo de treinamento é crucial para este tipo de técnica, e a qualidade desta etapa está diretamente relacionada com o desempenho dos classificadores. No treinamento são apresentados para o algoritmo exemplos de padrões rotulados para que sejam estabelecidas relações entre as características presentes em cada padrão e suas classes correspondentes (XU; LI; WANG, 2017).

Um exemplo de aprendizado supervisionado pode ser observado analisando o problema de classificação abordado nesta dissertação, o treinamento é feito fornecendo para os algoritmos, denominados classificadores, um conjunto de dados contendo uma série notícias rotuladas, indicando se as mesmas são verdadeiras ou falsas, os classificadores, cada um usando de suas estratégias, tentam identificar características presentes nas notícias verdadeiras e falsas, associando estas características aos rótulos indicados. A etapa seguinte é a classificação, fase em que são fornecidos aos classificadores, já treinados, um conjunto de dados não rotulado, denominado conjunto de teste, contendo notícias que não foram usadas no processo de treinamento, e então cada classificador tenta classificar este novo conjunto de dados de acordo com as relações entre características e rótulos estabelecidas no processo de treinamento (KOTSIANTIS, 2007).

3.2 MÉTODOS DE REPRESENTAÇÃO DE TEXTOS

Natural Language Processing (NLP) é a subárea da Inteligência Artificial que estuda a capacidade e as limitações de uma máquina em entender a linguagem dos seres humanos. Antes que um algoritmo de aprendizagem de máquina consiga classificar um texto, é ne-

cessário realizar alguns procedimentos. Uma declaração recebida em texto bruto, da forma que humanos conseguem ler e entender, é considerada um texto em linguagem natural, assim sendo, os textos em linguagem natural precisam passar por um processamento que visa identificar e extrair características transformando estes em uma representação estruturada, convertendo portanto o texto em um vetor de características, e só então, estas representações podem ser usadas para alimentar um algoritmo classificador. É muito importante que o vetor de características resultante deste processo conserve as informações originais contidas no texto, disponibilizando uma representação numérica a ser interpretada pelos algoritmos de classificação. Este trabalho utiliza cinco técnicas diferentes de extração de características que podem ser divididas em dois grupos: CountVectorizer e TF-IDF integrando o grupo *Bag-of-Words* (BoW) e os extratores Word2Vec, GloVe e FastText compondo o grupo *Word Embedding*.

3.2.1 Pré-processamento de textos

Uma boa prática antes do processo de extração de características é realizar uma limpeza no texto, removendo palavras sem grande valor semântico e que produzem ruídos no funcionamento dos classificadores. Estas palavras, usadas com muita frequência e que de modo geral não contribuem para o reconhecimento de padrões, são conhecidas como *stop words*. Existem listas predefinidas de *stop words*, alguns exemplos comuns na língua inglesa são: *"the", "a", "be", "if", "and"* (WILBUR; SIROTKIN, 1992). Após este processamento inicial, o texto é entregue para um extrator de características que realiza a sua conversão em um vetor.

3.2.2 Bag-of-Words

Os modelos baseados em *Bag-of-Words* fornecem uma maneira simples mas bastante eficiente de representação textual. Nestes modelos a ordem e a estrutura gramatical das frases não são importantes, de modo que, um texto é representado como uma coleção não ordenada de suas palavras.

3.2.2.1 CountVectorizer

CountVectorizer (CV) é uma implementação do modelo BoW, que recebem um dataset, S, e retorna uma matriz esparsa A de tamanho m por n, sendo que m é igual ao número total de padrões, e n é igual ao número total de palavras distintas usadas em S. A Matriz 3.1 é a representação hipotética de um conjunto de dados processado por um modelo CountVectorizer, cada linha representa um padrão, e cada $a_{i,j}$, representante de uma palavra do conjunto S, recebe como valor o número de vezes que está palavra aparece no

padrão (VIJAYARAGHAVAN et al., 2020).

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix} \tag{3.1}$$

Apesar de sua implementação simples, o CountVectorizer ainda apresenta um bom desempenho em diversos problemas quando comparado com extratores de características mais recentes, superando em alguns casos as modernas redes neurais profundas (KULKARNI; SHIVANANDA, 2019).

3.2.2.2 TF-IDF

O *Term Frequency Inverse Document Frequency* (TFIDF) cria uma matriz de representação do texto semelhantemente a representação criada pelo CountVectorizer, no entanto, esta técnica atribui peso a cada palavra, levando em conta a frequência inversa de ocorrências da palavra no texto, de forma que, uma palavra tida como rara, com poucas ocorrências, tem uma capacidade maior de generalização, portanto, um peso maior que uma palavra com várias ocorrências. O peso de cada palavra é definido por duas variáveis: a frequência de termo, e a frequência inversa de documentos (SALTON; BUCKLEY, 1988).

Para a aplicação da técnica, inicialmente é montado um vocabulário, Equação 3.2, que recebe como entrada o documento d .

$$V(d) = \sum_{t} n(t, d) \tag{3.2}$$

O próximo passo é o cálculo do *term frequency*, Equação 3.3, que indica a frequência que o termo $t \in V(d)$ aparece no documento d, onde $n(t, d)$ indica o número de ocorrência da palavra t no documento d.

$$tf(t, d) = \frac{n(t, d)}{V(d)} \tag{3.3}$$

Dada um conjunto de documentos D, o *inverse document frequency*, Equação 3.4, é o *log* do número de documentos N dividido por $df(t, D)$, o número de documentos $d \in D$ contendo o termo t. Como resultado, palavras frequentes em D terão um baixo *term frequency score*

$$idf(t, D) = \log\left(\frac{N}{df(t, D)}\right) \tag{3.4}$$

Finalmente, pode-se obter o cálculo do TF-IDF para cada palavra w, Equação 3.5, multiplicando o *term frequency* pelo *inverse document frequency* (VIJAYARAGHAVAN et al., 2020).

$$w(t,d) = tf(t,d) \times idf(t,D) \tag{3.5}$$

.

3.2.3 Word Embedding

Modelos baseados em *Word Embedding* realizam uma conversão de um conjunto de palavras ou frases em vetores numéricos, densos, de tamanho fixo. Diferente dos modelos baseados em BoW, técnicas de *Word Embedding* são capazes de armazenar informações sobre o contexto e valor semântico das orações, portanto, para estes modelos a ordem das palavras dentro da frase é importante. Cada palavra dentro da oração é representada por um ponto em um espaço multidimensional chamado *embedding space* que armazena os valores de cada dimensão/ponto referentes aquela palavra. Os valores destes pontos são definidos durante o processo de treinamento, no caso deste trabalho, de acordo com os modelos de *word embedding* pré-treinados referentes a cada técnica. Cada dimensão representa a posição do valor de cada palavra em relação a todas as outras, portanto a proximidade de duas palavras no *embedding space* representa a proximidade no valor semântico destas palavras (MANNING; SCHÜTZE, 1999) (LEBANON; MAO; DILLON, 2007) (LEVY; GOLDBERG, 2014).

3.2.3.1 Word2Vec

Desenvolvida pelo Google, o Word2Vec (W2V) é uma rede neural superficial de duas camadas que, recebendo como entrada um conjunto de palavras, tenta prever a próxima palavra reconstruindo seus contextos linguísticos. Consiste na combinação de duas técnicas: CBOW e Skip-gram, ambas as técnicas atribuem pesos e atuam como representações de vetores de palavras, a diferença é que enquanto o CBOW tenta prever a probabilidade de uma palavra em um contexto, levando em conta as palavras próximas, o Skip-gram inverte esta arquitetura, tentando prever o contexto dado uma palavra. Esta arquitetura foi revolucionária quando proposta, pois abre a possibilidade da preservação das analogias semânticas sob aritmética básica nos vetores de palavras, permitindo operações matemáticas com valores semânticos, por exemplo a equação: *king - man + woman = queen* (DROZD; GLADKOVA; MATSUOKA, 2016).

O Word2Vec atraiu muita atenção da comunidade científica desde sua apresentação, diversos trabalhos indicam a sua capacidade de gerar uma representação vetorial de alta qualidade, foi usado como base para a criação de outras abordagens, e com frequência é usado como parâmetro de comparação para novas propostas (RONG, 2014).

3.2.3.2 GloVe

Global Vectors for Word Representation, ou GloVe (GL) é uma rede neural de aprendizado não supervisionado, desenvolvido na universidade de Stanford que funciona de maneira semelhante ao Word2Vec, a diferença entre estas duas técnicas está na forma de contextualização das palavras. Enquanto o Word2Vec trabalha de maneira preditiva, o GloVe constrói uma matriz de coocorrência entre palavras e contexto, levando em conta a frequência que uma palavra aparece em um determinado contexto, o GloVe considera que a frequência de ocorrências é uma informação importante para o processo de treinamento, criando meios para que a combinação de vetores de palavras se relacione diretamente com a probabilidade de ocorrências dessas palavras no corpus(PENNINGTON; SOCHER; MANNING, 2014).

3.2.3.3 FastText

FastText (FT) é uma biblioteca desenvolvida pelo FaceBook, que combina alguns dos conceitos mais bem-sucedidos introduzidos pela comunidade acadêmica de processamento de linguagem natural. O que diferencia esta técnica das demais, é que em vez de considerar diretamente uma representação vetorial para uma palavra, como o Word2Vec e GloVe, o FastText considera uma representação para cada caractere n-gram, entendendo uma palavra como a soma dos caracteres n-gram que a compõe. Portanto, para o FastText, a menor unidade de treinamento é o n-gram, e não a palavra. Esta abordagem também faz uso do conceito de informações de sub-palavras e o compartilhamento de informações entre as classes por meio de uma representação oculta. Esta forma de organização gera melhores inserções para palavras raras e fora do corpus, fazendo com que o FastText se mostre eficiente em situações em que outras abordagens apresentam limitações (JOULIN et al., 2017; JOULIN et al., 2016).

3.3 SISTEMAS DE ENSEMBLE

Os Sistemas de Ensemble são meta-algoritmos que tem como base um único classificador, que por meio de combinações homogêneas, de diversas versões deste mesmo modelo, produz um modelo preditivo único, que tenta melhorar o desempenho reduzindo variações nos resultados (SYARIF et al., 2012; MARQUéS; GARCíA; SáNCHEZ, 2012).

3.3.1 Bagging

Bagging, também chamado de *Bootstrap Aggregating*, é uma técnica geralmente aplicada em algoritmos de árvore de decisão, mas nada impede seu uso em outros algoritmos de classificação. Consiste em dividir um mesmo dataset de treinamento em n partes, ou *bags*, de tamanhos iguais, podendo haver repetição de padrões em cada um dos subconjuntos, em

seguida, cada uma destas partes é utilizada no treinamento de uma versão de um mesmo classificador. A classificação final dos padrões de teste é definida pela combinação da resposta de todos os classificadores, usando a regra *averaging* para problemas de regressão ou *voting* para problemas de classificação (BREIMAN, 1996). Exemplos da implementação desta técnica utilizados neste trabalho são os classificadores RandomForest e Extra-Trees (BREIMAN, 2001; GEURTS; ERNST; WEHENKEL, 2006).

3.3.2 Boosting

A ideia por detrás desta técnica é, partindo de um classificador fraco, obter melhorias no desempenho treinando várias instâncias destes modelos em sequência, sendo que em cada treinamento, é dada ênfase nos padrões que o modelo anterior errou, buscando portanto, uma melhoria gradual no resultado por meio da compensação das fraquezas do modelo anterior. Este processo pode ser repetido diversas vezes, até que se obtenha um resultado satisfatório (SCHAPIRE, 1990; ZHOU, 2012; DIETTERICH, 2000). Um exemplo de implementação de *Boosting* experimentado neste trabalho é o AdaBoostClassifier (FREUND; SCHAPIRE, 1997).

3.4 MULTIPLE CLASSIFIER SYSTEMS

Multiple Classifier Systems (MCS) é uma solução que tem se mostrado eficiente para diversos tipos de problemas de classificação. Composta de um grupo de classificadores e uma função que determina a regra de combinação entre estes classificadores, é amplamente indicado na literatura que combinação de múltiplos classificadores podem oferecer vantagens sobre abordagens monolíticas (Tin Kam Ho; Hull; Srihari, 1994; ROLI, 2009; Fumera; Roli, 2005; HO, 2001). A ideia por detrás de se combinar mais de um algoritmo classificador para uma classificação conjunta deriva, da observação de que classificadores diferentes cometem padrões diferentes de erros, de modo que, uma classificação conjunta bem sucedida combina predominantemente os acertos individuais de cada um dos classificadores participantes de um grupo. O primeiro desafio em se combinar classificadores é encontrar um conjunto de classificadores que se complementam, de forma que cada classificador possa classificar corretamente os padrões classificados de forma errada pelos demais classificadores do grupo.

3.4.1 Regras de combinação

As técnicas básicas de combinação de classificadores empregadas neste trabalho são: Average, Product, Median, Maximum, Minimum, Voto Majoritário (Kuncheva, 2002; Ponti Jr., 2011). Nas subseções seguintes há uma breve explicação do funcionamento de cada uma destas técnicas. Para todas as definições a seguir, x é o vetor de entrada, i é a i-ésima classe e j é o j-ésimo classificador. $P_{ij}(x)$ é a probabilidade a posteriori obtido pelo j-ésimo

classificador para a i-ésima classe em relação à entrada x (CRUZ, 2011)(Kittler et al., 1998; Kuncheva, 2002).

3.4.1.1 Average

Na técnica **Average** cada classificador atribui uma probabilidade de que um padrão seja pertencente a uma das possíveis classes, em seguida, são somado a as probabilidades referentes a cada classe, e o padrão é classificado de acordo com a classe que for indicada como a mais provável.

3.4.1.2 Product

Conforme demonstrado na Equação 3.6, a combinação pela regra do **Product** define a classificação de um padrão de acordo com a distribuição de probabilidade conjunta do resultado da classificação de cada um dos *models* participantes da combinação.

$$C_{\mathrm{i}}(x) \sim \ argmax\left\{\prod_{j} P_{\mathrm{ij}}(x)\right\} \tag{3.6}$$

3.4.1.3 Median

A Median pode ser considerada uma melhoria da técnica Average, ela localiza a mediana das pontuações de cada classe entre os classificadores e atribui o padrão de entrada à classe com a pontuação máxima entre as medianas.

3.4.1.4 Maximum

Na regra **Maximum**, após cada classificador participante da combinação determinar uma probabilidade de um dado padrão pertencer a uma classe, a regra Maximum seleciona o classificador que apresenta a maior probabilidade posterior, ou seja, o classificador mais confiante em sua resposta.

3.4.1.5 Minimum

A estratégia **Minimum** de combinação recebe um padrão e seleciona a pontuação mínima de cada classe entre os classificadores da combinação, em seguida atribui o padrão de entrada a classe com a pontuação máxima entre as selecionadas.

3.4.1.6 Voto Majoritário

Na aplicação da técnica **Voto Majoritário**, cada um dos classificadores faz uma classificação individual de um padrão, a classificação final é aquela que receber maior número de votos, ou seja, a classe que o maior número de classificadores apontou.

3.5 ORACLE

Um importante conceito no estudo da viabilidade da combinação de classificadores é a ideia de *Oracle* (Kuncheva, 2002), que é representado por uma combinação ideal de classificadores, no qual são contabilizados apenas os acertos de cada classificador participante de uma dado conjunto. Desta forma é possível estabelecer: o limite de acertos que aquela combinação de classificadores em uma situação ideal pode atingir, verificar a diversidade do grupo de classificadores, além de possibilitar o descarte de combinações que não apresentam potencial de ganho. Caso o Oracle de um dado conjunto de classificadores seja igual ou próximo ao desempenho individual dos classificadores participantes do grupo, aquela combinação é considerada inviável, uma situação oposta, em que o resultado do Oracle seja superior ao resultado individual dos classificadores, uma combinação entre este grupo pode ser convertida em um ganho na classificação final.

Um grupo de classificadores que apresente comportamento similar entre seus integrantes não é considerado ideal para uma aplicação de técnicas de MCS, tendo em vista que, se todos os classificadores do grupo acertam e erram os mesmos padrões, a classificação em conjunto vai ter resultado equivalente a classificação monolítica. Portanto, quanto mais diverso for o comportamento dos classificadores do conjunto, mais interessante é este grupo para aplicação de técnicas de MCS.

3.5.1 Diversidade

No contexto de combinação de classificadores, o conceito de diversidade se refere a variação no padrão de acertos e erros de um determinado grupo de classificadores. Um grupo de classificadores que apresenta comportamento similar entre seus membros, não é um grupo ideal para aplicação de técnicas de combinação. Uma das formas de se verificar a diversidade de um conjunto de classificadores é por meio da análise do Oracle, quanto maior o Oracle em relação ao resultado das classificações individuais dos componentes do grupo, mais promissora se torna a aplicação de técnicas de combinação entre estes classificadores (Kuncheva; Whitaker, 2001; KUNCHEVA; WHITAKER, 2003; Tie-Gang Fan; Ying Zhu; Jun-Min Chen, 2008).

3.6 REDUÇÃO DE DIMENSIONALIDADE

Técnicas de redução de dimensionalidade fazem uso de uma uma base ou representação matemática dentro da qual, é possível representar a maioria, mas não todas as variações de dados presentes em uma matriz, mantendo na representação somente as informações mais relevantes, tornando possível, realizar a representação gráfica de matrizes de múltiplas dimensões. Sempre que se aplica uma técnica deste tipo, há uma perda de informações, sendo mantidas somente os dados consideradas mais representativos. A aplicação de técnicas deste tipo é utilizada em diversos problemas de *machine learning* e de análise de

dados, e existem diversas opções disponíveis para esta tarefa, algumas delas são: *Multidimensional Scaling* (MDS), *Principal Component Analysis* (PCA), *t-Distributed Stochastic Neighbor Embedding* (t-SNE) e *Uniform Manifold Approximation and Projection* (UMAP). Cada uma destas técnicas tem um funcionamento interno diferente, considerando partes distintas do conjunto de dados como prioritária, e o resultado final pode variar muito de acordo com a opção selecionada (KRUIGER et al., 2017; MCINNES; HEALY; MELVILLE, 2018; MAATEN; HINTON, 2008; COX; COX, 2008; WOLD; ESBENSEN; GELADI, 1987).

4 PROPOSTA

Analisando exclusivamente um classificador, é impossível avaliar a utilidade deste no conjunto que determina o processo de classificação combinada. Para entender a contribuição de cada classificador dentro do grupo, é fundamental entender a interação entre todos os candidatos à compor o *pool*, garantindo a diversidade do conjunto. A proposta apresentada nesta dissertação abre a possibilidade de identificar, de maneira visual, a interação entre um grupo de classificadores, eliminando candidatos que apresentam comportamento similar, e por tanto, não contribuem com a diversidade do conjunto.

Sabendo que um Oracle elevado não é garantia de uma melhora na classificação combinada, outra vantagem de ter uma visão panorâmica do comportamento de todos os classificadores, é a possibilidade de realizar uma substituição de um classificador do *pool*, sem prejuízo no Oracle, em resumo não basta garantir um Oracle elevado, é importante promover um encaixe entre *pool* e função de combinação. Supondo que um grupo de cinco classificadores apresente uma boa diversidade entre seus membros, mas a função de classificação não consiga tirar proveito dessa diversidade, de posse de um gráfico de comportamento dos classificadores, obtido por meio da aplicação da proposta, é possível saber quais classificadores podem ser retirados e incluídos no *pool* sem que esta substituição implique uma queda no valor do Oracle, processo que pode ser repetido até se encontrar o equilíbrio entre o *pool* e a função de combinação.

A Figura 1 mostra a arquitetura proposta neste trabalho, que é composta de sete módulos: extração de características, treinamento dos classificadores, classificação do conjunto de dados de validação, cálculo da matriz de dissimilaridade, redução de dimensionalidade, representação gráfica e seleção do grupo de classificadores. Como saída é obtido um subconjunto dos classificadores treinados.

Cada um dos M_i classificadores $\{M_1, M_2, \ldots, M_n\}$ é treinado com uma visão diferente do conjunto de dados de treinamento Γ, sendo que cada uma dessas visões é obtida usando um método de extração de características diferente. Em seguida, cada classificador realiza uma classificação monolítica do conjunto de dados de validação Θ, no resultado destas classificações é calculada a dissimilaridade, obtendo uma matriz multidimensional que representa o comportamento de cada classificador em relação a todos os outros que compõem o *pool*. Para que esta matriz com múltiplas dimensões possa ser representada graficamente, é aplicada uma técnica de redução de dimensionalidade que resulta em uma representação 2D da classificação. O gráfico resultante deste processo indica o comportamento de cada um dos classificadores dentro do grupo. Levando em conta a distância euclidiana entre os classificadores, é possível determinar a proximidade no seu comportamento, sendo que quanto mais próximo dois classificadores são representados no gráfico, mais similar é o resultado da classificação entre eles, e quanto mais distante é esta representação, mais

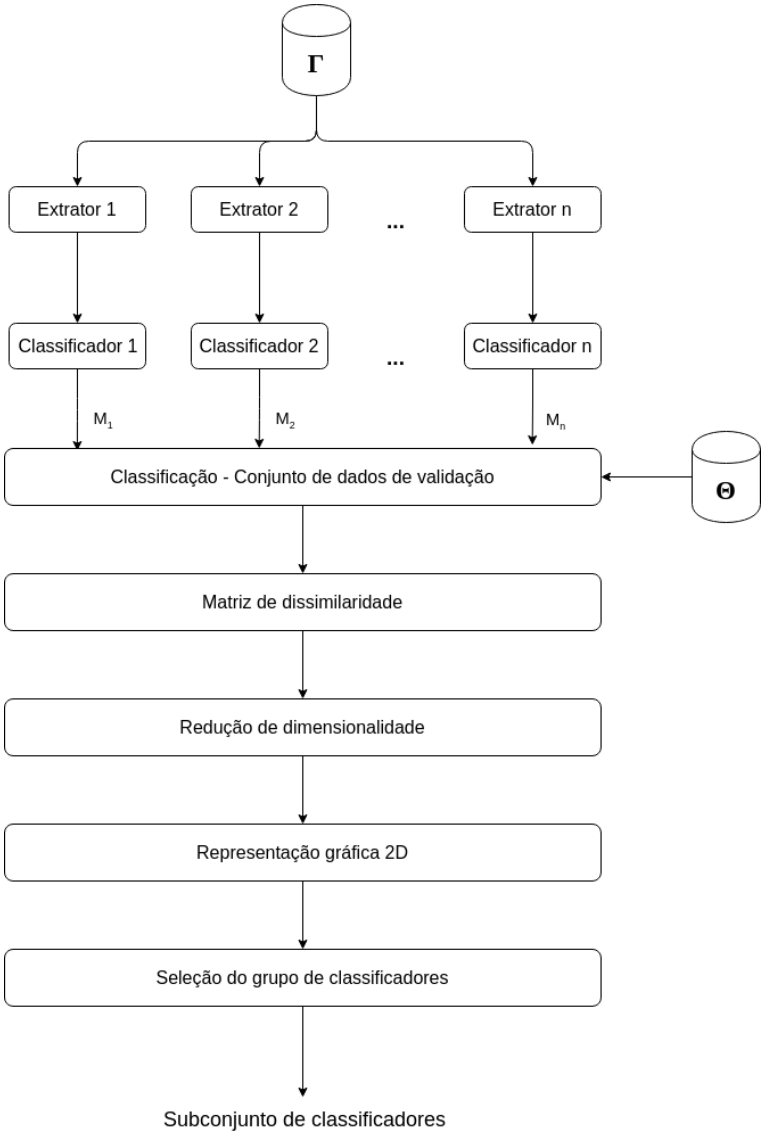


Figura 1 – Visão geral da metodologia proposta para selecionar os conjuntos de classificadores mais promissor dentro do *pool*. Γ representa o conjunto de dados de treinamento e Θ representa o conjunto de dados de validação.

variado é o padrão de acertos e erros entre estes dois classificadores(PEKALSKA; DUIN; SKURICHINA, 2002). Portanto, um grupo de classificadores representados por pontos mais afastados no gráfico apresentam um Oracle maior se comparado com classificadores representados por pontos mais próximos. O último módulo consiste na seleção de um grupo de classificadores que represente um bom potencial de ganho em uma classificação conjunta.

Os módulos matriz de dissimilaridade, redução de dimensionalidade e seleção de classificadores são descritos nas seções seguintes.

4.1 MATRIZ DE DISSIMILARIDADE

O resultado da classificação monolítica é usado como valor de entrada para uma função de cálculo de diversidade, que é responsável por retornar uma matriz simétrica multidimen-

sional, construída de forma que cada célula $d(i, j)$ representa a dissimilaridade entre os modelos M_i e M_j. Sabendo que a escolha desta função é crucial para a proposta da dissertação, uma análise cuidadosa foi feita na literatura, que apontou as técnicas *Q Statistic, Disagreement Mensure* e *Double-fault* como as mais indicadas para o problema (GAMA; SEBASTIãO; RODRIGUES, 2013; YULE, 1900; Tin Kam Ho, 1998; KUNCHEVA; WHITAKER, 2003; ROLI; KITTLER, 2002).

O *Disagreement Mensure* representa a probabilidade de que dois classificadores discordem, enquanto o *Double-fault* indica a probabilidade de ambos os classificadores tomarem uma decisão incorreta, ou seja, para o primeiro, o retorno de valores mais elevados indicam maior diversidade, enquanto para o segundo valores mais alto indicam menor diversidade no conjunto. Observando a descrição destas duas técnicas na literatura, e experimentando seu funcionamento nos conjuntos de dados analisados, foi possível concluir que ambas são equivalentes, diferindo apenas na forma final de representação dos dados. (PAL; PAL, 2002; SKALAK, 1996). Já o *Q Statistic* é comumente usado para avaliar o nível e de dependência entre dois classificadores, tendo sempre uma saída binária em que -1 significa dependência negativa total e 1 dependência positiva total. Esta característica causou uma distorção na saída dos dados quanto o *Q Statistic* foi experimentado no problema desta dissertação, de forma que, classificadores com comportamento semelhante, mas não idêntico, frequentemente foram representados por pontos sobrepostos (KUNCHEVA; WHITAKER, 2003).

Um fator determinante para a escolha da técnica foi o sucesso obtido em outras publicações que se propuseram a tratar problemas similares. Portanto, para este trabalho optou-se pelo uso do *Double Fault* (CAVALCANTI et al., 2016; CRUZ et al., 2013; GIACINTO; ROLI, 2001). A Equação 4.1 mostra o cálculo da dissimilaridade entre dois modelos (M_i e M_j) usando *Double Fault*:

$$d(i, j) = \frac{N^{00}}{N^{00} + N^{01} + N^{10} + N^{11}} \tag{4.1}$$

sendo N^{ij} o número de padrões classificados corretamente (1) ou o número de padrões classificados de maneira incorreta (0) por cada modelo M_i e M_j. Assim, entende-se que o *Double-Fault* mede a probabilidade de que o dois classificadores classifiquem incorretamente um dado padrão.

4.2 REDUÇÃO DE DIMENSIONALIDADE

Considerando a relevância desta etapa, diversas opções foram consideradas para a tarefa. Inicialmente foram observadas na literatura as principais abordagens para o problema proposto, são elas: t-SNE, UMAP, PCA e MDS. Experimentos publicados indicaram uma melhor precisão das técnicas t-SNE e UMAP em relação ao PCA e MDS (ESPADOTO et al., 2019; VERMA; SALISBURY, 2020). Uma ferramenta que auxiliou na confirmação

dos resultados apontados na literatura foi o *Embedding Projector* [1], plataforma em que foram realizados testes com os conjuntos de dados que, futuramente seriam analisados pela proposta desta dissertação. Em seguida, experimentos usando as mesmas entradas indicaram que as duas técnicas selecionadas, t-SNE e UMAP, se mostraram igualmente precisas para o problema abordado neste trabalho. Não havendo grandes discrepâncias entre os valores obtidos com as duas técnicas, e considerando igualmente satisfatórios os resultados, foi analisado o tempo de execução de cada abordagem. O UMAP teve uma execução consideravelmente mais rápida nas matrizes de até 25×25 dimensões, sendo superado pelo t-SNE conforme se acrescentavam novas dimensões na matriz.

Considerando a similaridade nos resultados entre as duas técnicas, e que a aplicação da proposta deste trabalho pode avaliar um números variados de classificadores, optou-se pelo uso da t-SNE.

4.3 SELEÇÃO DE CLASSIFICADORES

Aplicando a técnica de redução de dimensionalidade na matriz de dissimilaridade, obtêm-se como resultado uma matriz de duas dimensões, que representa a diferença de comportamento entre cada um dos classificadores do *pool*. Construindo uma representação gráfica desta matriz, e considerando que os classificadores devem apresentar comportamento similar na classificação do conjunto de dados de validação e teste, é possível prever qual subconjunto de classificadores apresentará o maior Oracle na classificação do dataset de teste, tornando possível portanto, selecionar os subconjunto mais promissores para aplicação de técnicas de MCS. A seleção do grupo final de classificadores é feita de maneira visual, considerando a distância euclidiana entre cada ponto do gráfico. Caso o resultado obtido com uma determinada configuração do *pool* seja inferior ao esperado, a proposta abre a possibilidade da substituição de um integrante do conjunto sem prejuízo do Oracle.

4.4 VANTAGENS E LIMITAÇÕES DA PROPOSTA

A identificação de quais são os melhores classificadores para a composição do *pool* que resulta em uma melhor aderência à função de classificação não é uma tarefa trivial, e como discutido anteriormente, é fator determinante no sucesso da implementação da classificação combinada. Diversas publicações demonstram que a busca por um conjunto adequado, quando feita sem uma ferramenta de auxílio, é um processo exaustivo e que requer diversos testes, exigindo um consumo excessivo de recursos humanos e computacionais(BROWN; KUNCHEVA, 2010; Coelho; Von Zuben, 2006). Uma vantagem da aplicação da proposta aqui apresentada é abreviar este processo, mantendo sempre uma alta diversidade no *pool* e evitando testes com conjuntos que apresente baixa possibilidade de melhoria no desempenho. No entanto, conforme discutido nas seções anteriores, técnicas de combinações de

[1] http://projector.tensorflow.org/

classificadores bem sucedidas consistem em uma junção entre um *pool* de classificadores, que deve apresentar uma grande diversidade entre seus membros, e uma função de combinação, que deve ser capaz de tirar proveito da diversidade do *pool*. Conjuntos de classificadores que apresentem um Oracle elevado, não garantem a melhoria na classificação final, podendo apresentar resultados variados a depender da função de combinação selecionada. No entanto, um conjunto com Oracle baixo sempre vai apresentar resultados baixos na classificação final, independente da função utilizada. Este trabalho se limita a apresentar a interação entre um grupo de classificadores, indicando qual composição de *pool* apresentam maior diversidade entre seus membros e quais classificadores de um conjunto podem ser substituídos sem prejuízo do Oracle.

5 EXPERIMENTO

Neste capítulo são apresentados os experimentos realizados para a validação do método proposto. A Seção 5.1 apresenta os conjuntos de dados utilizado. As Seções 5.2 e 5.3 apresentam, respectivamente, as configurações do experimento e a métrica de avaliação. As Seções 5.4, 5.5 e 5.6 apresentam os resultados dos experimentos.

5.1 CONJUNTO DE DADOS

Para validação da proposta apresentada nesta dissertação, foram realizados experimentos com dois conjuntos de dados diferentes: o Liar Dataset, apresentado na Seção 5.1.1 e o FA-KES Dataset, apresentado na Seção 5.1.2.

5.1.1 Liar dataset

Um dos conjuntos de dados selecionado para a realização de experimentos neste trabalho é o LIAR Dataset (WANG, 2017), construído usando avaliações do site PolitiFact.com. Ele contém declarações tratando de temas relacionados à política americana, retiradas de debates, de entrevistas, de comunicados de imprensa, de tweets, entre outros.

Esse dataset é composto por 12.836 padrões que estão divididos em seis classes conforme mostrado na Tabela 3. Cada um dos padrões é representado pelos seguintes atributos: *label, statement, subject, speaker, job, state, party* e *venue*. A Tabela 2 apresenta o exemplo de um padrão presente no conjunto de dados.

statement	There are more people killed with baseball bats and hammers than are killed with guns.
subject	guns
speaker	paul-broun
job	Congressman
state	Georgia
party	republican
label	pants-fire

Tabela 2 – Exemplo de padrão presente no Liar Dataset

Para ampliar o alcance do experimento, foi feito uma divisão binária do dataset, considerando os padrões rotulados como *true, mostly-true e half-true* como ***true***, ou seja, não são considerados *fake news*, enquanto os padrões rotulados como *barely-true, false e*

	pants-fire	false	barely-true	half-true	mostly-true	true	Total
Train	839	1995	1654	2114	1962	1676	10240
Valid	116	263	237	248	251	169	1284
Test	92	249	212	265	241	208	1267

Tabela 3 – Número de padrões Liar Dataset

pants-fire são considerados ***false***, ou seja, *fake news*. A Tabela 4 mostra a quantidade de padrões por classe após a agregação das classes originais do banco de dados.

	true	false	Total
Train	5752	4488	10240
Valid	668	616	1284
Test	714	553	1267

Tabela 4 – Número de padrões Liar Dataset considerando uma divisão binária de classes

A escolha deste conjunto de dados se deu devido à sua popularidade entre a comunidade científica que se dedica a pesquisa deste tema, possibilitando assim a comparação dos resultados do experimento aqui apresentado com outros trabalhos já publicados.

5.1.2 FA-KES Dataset

O FA-KES Dataset é um conjunto de dados de *fake news*, em língua inglesa, com notícias coletadas de diversas fontes relacionadas a guerra na Síria. Estudos indicam a existência de uma forte correlação positiva entre o compartilhamento de uma *fake news* e as opiniões de quem às compartilha. Esta relação entre notícia e opinião é fonte de subjetividade no processo de classificação de uma informação por humanos, dificultando muito a construção de um conjunto de dados (VOSOUGHI; ROY; ARAL, 2018). Para lidar com este tipo de problemas, o FA-KES foi construído usando um processo semi-supervisionado de rotulagem, buscando informações contidas no banco de dados do *Violation Documentation Center* (VDC), e estabelecendo critérios objetivos para reduzir a interferência humana no processo. Este conjunto de dados consiste em um único arquivo contendo 804 artigos, sendo que 426 destes são rotulados como *True* e 378 são rotulados como *False* (SALEM et al., 2019). Cada notícia presente no FA-KES é formado pelos campos: *id, title, content, source, date, location e labels*, sendo que o campo *content* concentra a maior quantidade de informações, contando com em média 670 palavras por padrão. A Tabela 5 apresenta um exemplo de padrão contido no dataset.

ID	1965511221
Title	Turkish Bombardment Kills 20 Civilians in Syria
Content	Turkish shelling and air strikes killed at least 20 civilians...
Source	manar
date	8/28/2016
location	aleppo
label	True

Tabela 5 – Exemplo de padrão presente no FA-KES dataset.

5.2 CONFIGURAÇÕES DO EXPERIMENTO

Com o intuito de analisar um grande número de possibilidades de combinações, e avaliar o comportamento de classificadores de diferentes paradigmas (LANGLEY; SIMON, 1995), foram selecionados 16 máquinas de aprendizado, grupo composto pelos classificadores que apresentaram os melhores resultados em outros experimentos que utilizam os mesmos conjuntos de dados, adicionando ainda classificadores que não foram experimentados na literatura.

Como o discutido anteriormente, a composição do *pool* é fundamental para o sucesso de uma combinação de classificadores, levando em conta ainda a baixa acurácia apresentada em outras publicações que fazem uso dos mesmos conjuntos de dados, discutida no Capítulo 2, é interessante a análise do comportamento de outros classificadores ainda não experimentados neste problema. Com o intuito de encontrar a melhor configuração possível, além de entender o comportamento de técnicas distintas de classificação, foram selecionados os classificadores que apresentaram melhor desempenho em publicações que fizeram experimentos com os mesmos conjuntos de dados, e adicionados outros classificadores que não foram experimentados em nenhuma publicação verificada. O grupo de classificadores selecionados para o experimento são: AdaBoost (ADB), BernoulliNB (BNB), ComplementNB (CNB), ExtraTrees (ET), GaussianNB (GNB), K Nearest Neighbor (KNN), LogisticRegression (LR), MultinomialNB (MNB), Passive Aggressive (PA), RandomForest (RF), Stochastic Gradient Descent (SGD), Support Vector Machine - Linear SVC (SVM) e as redes neurais Multi-layer Perceptron (MLP), Perceptron (P), CNN e LSTM . Os extratores de características são: CountVectorizer (CV), *Term Frequency Inverse Document Frequency* (TFIDF), Word2Vec (W2V), Glove (GL) e FastText (FT). Python foi a linguagem usada para a implementação e avaliação dos modelos (GERS, 1999; KALCHBRENNER; GREFENSTETTE; BLUNSOM, 2014; RASCHKA; MIRJALILI, 2019; JOULIN et al., 2016; PENNINGTON; SOCHER; MANNING, 2014; RONG, 2014) .

Considerando as combinações entre classificadores e extratores de características, são avaliados ao todo 80 conjuntos. Os hiperparâmetros de cada classificador foram ajustados usando *GridSearchCV*, ferramenta que recebe como entrada um conjunto de dados de

treinamento, um classificador e uma lista de hiperparâmetros referentes ao classificador, e fornece como saída os melhores parâmetros para o modelo sob investigação, estes ajustes melhoram o desempenho da classificação individual, de forma a ajustar um classificador genérico para um problema específico, os hiperparâmetros ajustados são apresentados nas Tabelas 19, 20 e 21.

Foi demonstrado em outros trabalhos que usam o Liar Dataset, que alguns classificadores, quando recebem como valor de entrada o *statement* em conjunto com outros atributos disponíveis no dataset, têm tendência a melhorar o resultado da classificação, variando de acordo com os atributos adicionados. No entanto, diferentes trabalhos obtiveram seus melhores resultados adicionando diferentes atributos como valor de entrada para diferentes classificadores. Contudo, é uma constante em todos os trabalhos analisados o teste dos classificadores usando como valor de entrada somente o *statement*. Este comportamento fica evidenciado por meio de uma análise comparativa entre os trabalhos apresentados por Wang 2017 (WANG, 2017) e por Alhindi 2018 (ALHINDI; PETRIDIS; MURESAN, 2018). Ambos conseguem melhorar o desempenho dos classificadores adicionando os demais campos do dataset como valor de entrada para treinamento dos mesmos, no entanto, cada trabalho apresenta seu melhor resultado adicionando campos diferentes. Considerando que este trabalho tem foco na classificação em grupo, para estabelecer uma comparação justa entre os resultados obtidos nos nossos experimentos com outras publicações que fazem uso do mesmo conjunto de dados, colorblack além de assegurar que qualquer possível melhora na classificação seja obtida devido ao uso de técnicas de combinação dos classificadores, e não pela disponibilização de mais dados para treinamento, optou-se por usar como valor de entrada dos classificadores somente o *statement*.

5.2.1 Extração de características

Para os extratores do grupo *Bag-of-Words* (CV e TFIDF) o treinamento foi feito com o conjunto de dados de treinamentos do Liar Dataset, que resultou em uma matriz esparsa de 10.240×11.915, que de acordo com a explicação apresentada na Seção 3.2.2, representam respectivamente a quantidade de padrões e a quantidade de palavras distintas encontradas no conjunto de dados de treinamento, portanto, neste caso, um padrão é formado por um vetor de 11.915 características. Esta é a entrada fornecida aos classificadores que fazem uso destes extratores de características.

O treinamento dos extratores de características do grupo *Word Embedding* (W2V, GL e FT) foi feito com um pacote de palavras pré-treinado composto por um milhão de palavras em inglês, retiradas da Wikipédia (MIKOLOV et al., 2018). De acordo com a literatura, a qualidade das representações vetoriais melhora à medida que se aumenta o tamanho do vetor até atingir 300 dimensões, após este ponto, a qualidade dos vetores tende a estabilizar, e em alguns casos, até mesmo diminuir, variando de acordo com o número de palavras representadas em um padrão (Das et al., 2019; PENNINGTON; SOCHER;

MANNING, 2014). Por esta razão, considerando uma média de 18 palavras por padrão no conjunto de dados analisado, optou-se neste trabalho por usar um vetor de 300 dimensões por padrão, resultando em um vetor de características de treinamento de 10.240×300, que representam respectivamente a quantidade de padrões e o número de dimensões do vetor. Portanto, neste caso, cada classificador que faz uso destes extratores de características, recebe como entrada uma representação de cada padrão composta por um vetor de 300 características, montado de acordo com o pacote de palavras pré-treinado.

5.3 MÉTRICAS DE AVALIAÇÃO

É possível observar que em algumas publicações que utilizam o Liar Dataset nos experimentos, o uso da acurácia como métrica de desempenho dos classificadores (WANG, 2017), em outros os autores prefere fazer uso do F1-score (ALHINDI; PETRIDIS; MURESAN, 2018), e ainda assim, comparam seus resultados livremente sem nenhum tipo de conversão, pois de acordo com a literatura, em um conjunto de dados balanceado as duas métricas são equivalentes (GOUTTE; GAUSSIER, 2005). Em testes realizados neste trabalho observou-se que a variação entre as duas métricas de desempenho no conjunto de dados analisado acontece somente a partir da terceira casa decimal, tornando portanto possível a livre comparação entre publicações que fazem uso de qualquer uma das duas métricas de desempenho.

Neste trabalho optou-se pelo uso da acurácia como métrica de avaliação de desempenho dos classificadores, que pode ser obtida somando os padrões classificados corretamente e dividido pelo número total de padrões avaliados, conforme demonstrado na Equação 5.1:

$$Acc = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{5.1}$$

sendo TP a quantidade de padrões verdadeiros positivos, FP são os falsos positivos, TN são os verdadeiros negativos e FN são os falsos negativos.

5.4 CLASSIFICAÇÃO BINÁRIA LIAR DATASET

Com o objetivo de se estabelecer um *baseline*, foi feita uma classificação binária do Liar Dataset, os resultados apresentados na Tabela 6 são a acurácia desta classificação. Cada máquina de aprendizado, composta por classificador + extrator de características, foi treinada com o conjunto de dados de treinamento, os hiper-parâmestros utilizados em cada classificador, e o *range* utilizados pelo gridsearchcv são apresentados na Tabela 19. O melhor resultado absoluto foi obtido usando LogisticRegression com os dados representados pelo método FastText que alcançou 62,00% de acerto. Outros classificadores conseguiram uma taxa superior a 61,00%, tais como: LSTM+CV, MultinomialNB+CV e Passive Aggressive+Word2Vec. O melhor resultado de cada classificador está destacado em negrito.

Técnica	CV	TFIDF	W2V	FastText	GloVe
AdaBoostClassifier	59,40	**60,20**	58,80	57,00	57,70
BernoulliNB	**60,10**	60,00	59,10	58,10	58,20
CNN	**61,88**	59,35	57,46	61,17	57,46
ComplementNB	62,00	**62,10**	59,90	58,20	55,60
ExtraTrees	58,80	59,90	59,70	**60,10**	57,90
GaussianNB	48,60	49,90	57,60	**59,00**	57,20
KNN	57,10	**60,80**	59,40	57,90	56,90
LSTM	**61,48**	59,83	60,30	60,54	57,54
LogisticRegression	59,90	61,60	61,20	**62,00**	59,70
MLP	57,50	56,70	**59,80**	58,10	57,70
MultinomialNB	**61,60**	61,30	56,10	56,70	56,40
Passive Aggressive	56,90	60,50	**61,40**	60,00	49,70
Perceptron	**58,60**	56,60	43,50	43,80	56,10
RandomForest	58,40	59,70	60,40	**61,30**	54,40
SGD	61,20	**61,70**	61,60	61,40	53,40
SVM	57,90	60,20	61,50	**61,90**	59,70

Tabela 6 – Acurácia dos modelos usando diferentes métodos de extração de características considerando a classificação binária. O melhor resultado por modelo está em negrito.

5.4.1 Oracle

Como discutido na Seção 3.5, o Oracle pode ser usado para verificar a viabilidade de uma combinação de classificadores. Foram avaliados 17 conjuntos de classificadores, 16 compostos por cinco versões de um mesmo classificador, variando apenas o extrator de características, e um conjunto composto por todos os 80 classificadores participantes do experimento. A Tabela 7 mostra a acurácia de cada uma das 17 combinações.

Comparando os resultados da Tabela 6 com os da Tabela 7, verifica-se que existe uma margem significativa de possível melhoria em termos de precisão caso os classificadores sejam combinados de maneira eficiente. O melhor resultado na Tabela 6 foi de 62,9% de acerto, bem inferior ao melhor resultado apresentado na Tabela 7 que atingiu cem por cento de acerto.

5.4.2 Seleção de classificadores

Como discutido no Capítulo 4, a seleção de um grupo adequado de classificadores é o primeiro passo para uma classificação em conjunto bem sucedida. Com este propósito foi aplicada a técnica apresentada no Capítulo 4 que resultou no gráfico mostrado na Figura 2, que representa de maneira visual o comportamento de cada classificador.

Técnica	Nº de Acertos	Acurácia (%)
AdaBoostClassifier	1114	87,92
BernoulliNB	1078	85,08
CNN	1115	88,00
ComplementNB	1079	85,16
ExtraTrees	1078	85,08
GaussianNB	1118	88,24
KNN	1133	89,42
LSTM	1115	88,00
LogisticRegression	1089	85,95
MLP	1148	90,61
MultinomialNB	904	71,35
Passive Aggressive	1196	94,40
Perceptron	1262	99,61
RandomForest	1125	88,79
SGD	1143	90,21
SVM	1103	87,06
Todos	1267	100,00

Tabela 7 – Oracle por conjunto de classificadores composto por cinco versões de um mesmo classificador, variando apenas o método de extração de características(CountVectorizer, TfIdf, Word2Vec, Glove e FastText) entendendo o problema como binário.

5.4.2.1 Análise de comportamento

Para verificar a acurácia da técnica proposta, foram selecionados dois conjuntos de classificadores: Conjunto 1 e Conjunto 2, que tem seus membros representados respectivamente por pontos próximos (BNB+CV, BNB+TFIDF, CNB+TFIDF) e distantes (MNB+W2V, P+FT, SVM+TFIDF) na Figura 2. O diagrama Venn, Figura 3, mostra a diferença nos padrões classificados corretamente pelos dois grupos.

É possível observar que os classificadores do Conjunto 1 apresentam comportamento similar entre seus membros de forma que 695 padrões foram classificados corretamente por todos os três classificadores. Considerando apenas os classificadores BNB+CV e BNB+TFIDF, que apresentaram pontos sobrepostos na Figura 2, é possível observar que os mesmos apresentam comportamento idêntico na classificação, portanto, não há possibilidade de que uma combinação entre estas duas máquinas de aprendizado consiga uma melhora de desempenho em relação à classificação monolítica.

Um comportamento oposto é observado no Conjunto 2 (Figura 3), no qual 7 padrões foram classificados corretamente por todos os três classificadores, mas os padrões de acertos individuais variam muito, tornando portanto este um conjunto promissor para

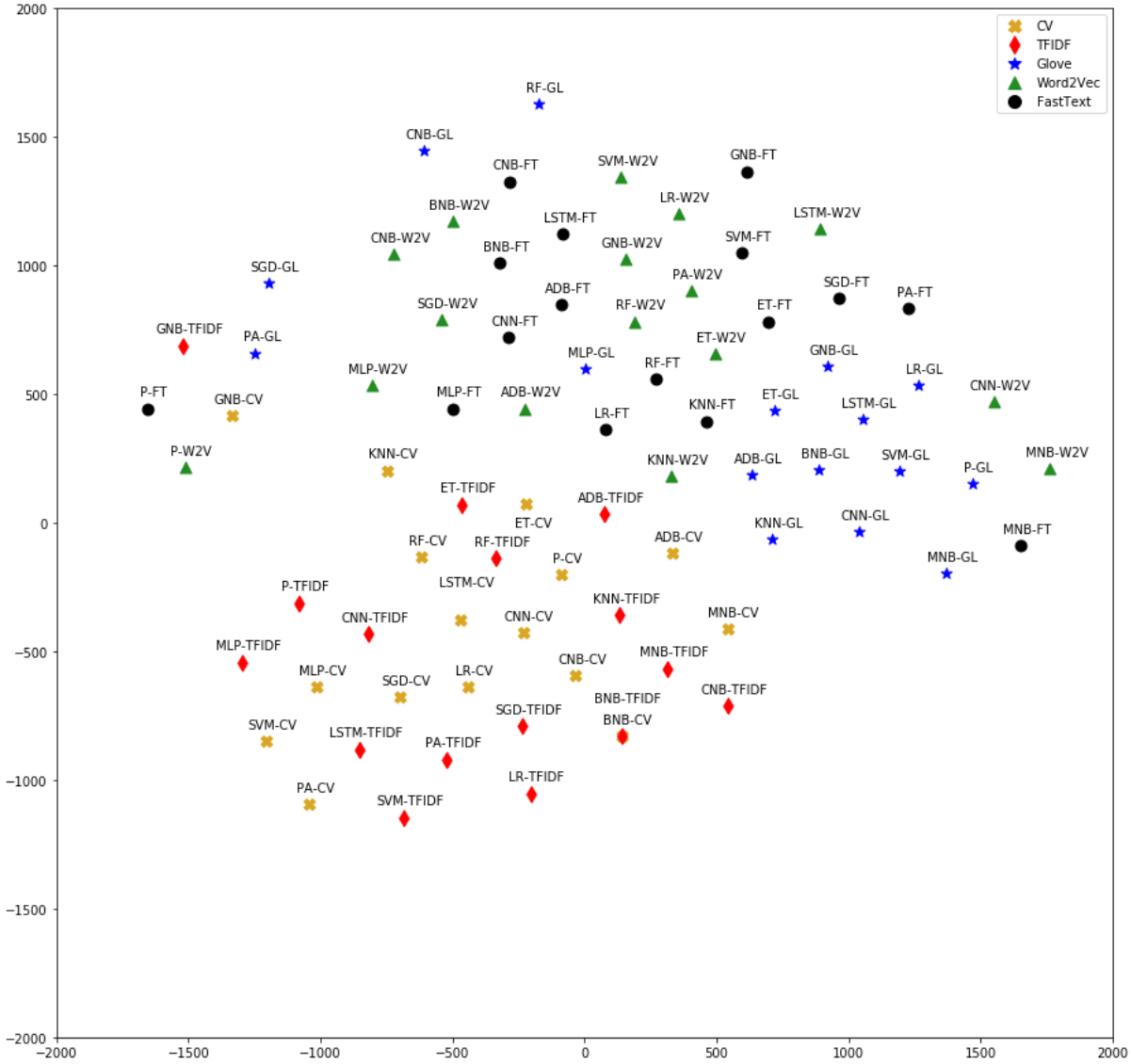


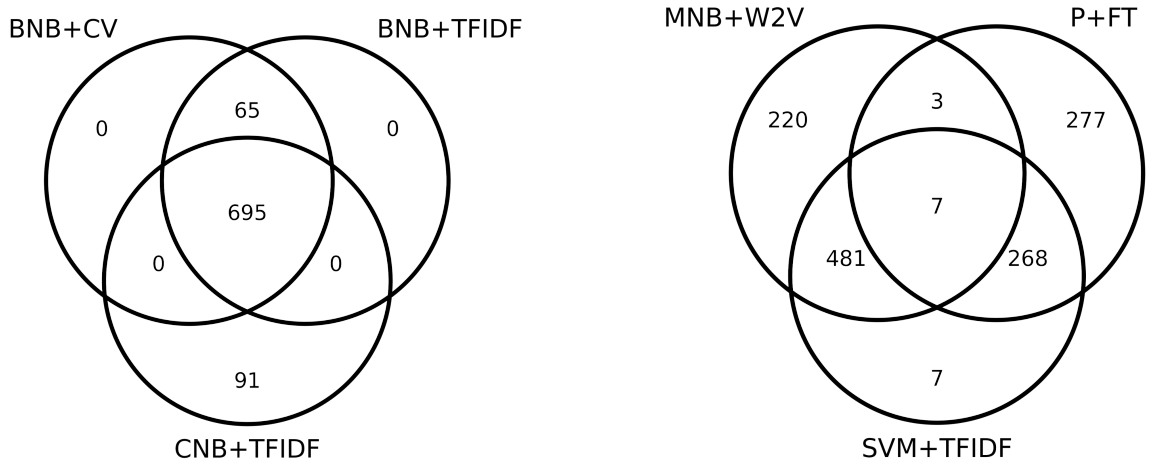
Figura 2 – t-SNE contendo todos os models utilizados neste trabalho considerando uma divisão binária de classes

aplicação de técnicas de MCS. A diversidade do comportamento entre estes dois conjuntos pode ser observada também analisando o Oracle de cada grupo (Conjunto 1: 851 acertos, Oracle=67,16% | Conjunto 2: 1263 acertos, Oracle=99,68%).

5.4.3 Aplicação de técnicas de MCS

É possível concluir que uma menor dispersão nos pontos do diagrama t-SNE (Figura 2) se traduz em um Oracle menor, significando que os modelos envolvidos na combinação apresentam comportamento similar. Logo, uma possível combinação desses classificadores não seria tão promissora quanto uma combinação entre um conjunto que apresente pontos mais esparsos no diagrama.

Considerando os resultados do Oracle, que indicam uma possível melhoria de desem-



(a) Conjunto 1 (b) Conjunto 2

Figura 3 – Análise de comportamento de dois conjuntos de classificadores. Conjunto 1 composto por elementos próximos e Conjunto 2 composto por elementos mais distantes do ponto de vista do t-SNE da Figura 2.

penho na classificação conjunta em relação a monolítica, foram aplicadas as técnicas de MCS apresentadas na Seção 3.4.1 nos mesmos conjuntos testados na Tabela 7.

A Tabela 8 mostra os resultados de regras de combinação (voto majoritário, *Average*, *Product*, *Median*, *Min* e *Max*) na qual cada linha é formada por um *pool* homogêneo com cinco classificadores; cada classificador é treinado com um dos cinco métodos de extração de características: CV, TFIDF, W2V, FastText e GloVe.

As saídas padrões dos classificadores Passive Aggressive, SVM-Lineare e Perceptron não possuem a opção de predição por probabilidade, logo, as combinações por Average, Maximum, Minimum, Median e Product, não foram calculadas; apenas a combinação por voto majoritário é apresentada para esses classificadores, as outras colunas foram preenchidas com "-". De maneira geral, combinar os *pools* homogêneos de classificadores não apresentou vantagem quando comparado aos resultados individuais apresentados na Tabela 6. Destaque para o *pool* que usou o SGD como classificador base e alcançou 63,10% de acerto.

5.4.4 Resultados

Sabendo identificar quais subgrupos dentro dos 80 classificadores experimentados neste trabalho apresentam melhor possibilidade de ganho em uma classificação conjunta, foi feita uma série de combinações usando os modelos que apresentaram pontos mais distantes entre si na Figura 2, e as técnicas de combinações apresentadas na Seção 3.4.1. O melhor resultado foi obtido com um grupo formado pelos classificadores: MNB+GL, SGD+CV, RF+W2V, KNN+W2V, usando a técnica de combinação Average, alcançando uma taxa de acerto de 64,70%. Desempenho superior a resultados apresentados na lite-

Técnica	Average	Max	Median	Min	Product	Voto Majoritário
AdaBoostClassifier	60,50	61,10	60,90	61,10	60,50	60,90
BernoulliNB	60,20	59,10	60,40	59,10	59,80	60,40
CNN	59,40	60,66	59,80	61,60	58,77	60,90
ComplementNB	60,90	61,60	60,90	61,60	60,90	60,90
ExtraTrees	61,80	61,40	60,60	61,40	61,60	61,90
GaussianNB	58,20	48,30	58,50	48,50	48,50	58,50
KNN	60,50	59,70	60,30	59,70	60,90	60,30
LSTM	60,00	60,40	57,60	61,40	60,60	60,90
LogisticRegression	61,60	61,50	61,30	61,50	61,60	61,30
MLP	59,50	57,90	59,50	57,90	58,70	59,50
MultinomialNB	60,40	61,20	56,70	61,20	60,50	56,70
Passive Aggressive	-	-	-	-	-	62,00
Perceptron	-	-	-	-	-	55,30
RandomForest	60,40	59,30	62,00	59,30	60,40	60,90
SGD	62,40	61,60	63,10	61,60	62,40	63,10
SVM	-	-	-	-	-	62,20
Todos	62,20	48,30	62,00	48,50	48,50	60.19

Tabela 8 – Taxa de acerto considerando o problema binário (%) de combinação de um mesmo classificador fazendo uso dos extratores de características: CountVectorizer, TfIdf, Word2Vec, Glove e FastText.

ratura (Tabela 9) que usam apenas o *statement* como valor de entrada e que tratam o LIAR dataset como um problema binário.

Método	Taxa de acerto (%)
Rashkin et al. (RASHKIN et al., 2017)	56,00
Khan et al. (KHAN et al., 2019)	60,00
Alhindi et al. (ALHINDI; PETRIDIS; MURESAN, 2018)	61,00
Elhadad, Mohamed K. et al. (ELHADAD; LI; GEBALI, 2019)	62.00
Combinação Média (5 SGD)	63,10
Proposta (4 classificadores + Average)	64,70

Tabela 9 – Comparação dos resultados apresentados neste trabalho com artigos que realizaram experimentos similares

Vale ressaltar ainda que na Tabela 6 é mostrado o resultado da combinação usando 5 versões do classificador SGD, cada um fazendo uso de um extrator de características diferente, atingindo uma acurácia de 63,10%, resultado superior que o apresentado na literatura.

5.5 CLASSIFICAÇÃO COM MÚLTIPLAS CLASSES LIAR DATASET

Como sequência do experimento, foram repetidas as aplicações de todas as técnicas apresentadas na proposta deste trabalho, considerando os rótulos originais de seis classes do Liar Dataset, apresentada na Tabela 3. Cada classificador foi treinado com o conjunto de dados de treinamento, em seguida foi feita uma classificação monolítica do conjunto de dados de teste,os hiper-parâmestros utilizados em cada classificador, e o *range* utilizados pelo gridsearchcv são apresentados na Tabela 20, os resultados desta classificação são apresentados na Tabela 10.

Técnica	**CV**	**TFIDF**	**W2V**	**FastText**	**GloVe**
AdaBoostClassifier	**24,1**	21,7	21,8	23,0	20,7
BernoulliNB	**25,0**	**25,0**	24,0	22,7	23,8
CNN	24,8	**25,1**	24,1	22,9	21,8
ComplementNB	23,4	**24,8**	23,6	22,8	20,6
ExtraTrees	**26,4**	24,9	24,7	22,0	20,7
GaussianNB	17,9	17,6	**23,4**	22,4	21,8
KNN	22,1	**23,0**	22,8	21,7	20,1
LSTM	22,0	**23,0**	21,8	21,3	21,1
LogisticRegression	23,5	**25,0**	24,1	24,8	22,5
MLP	23,1	**23,8**	22,7	23,4	21,8
MultinomialNB	22,3	**23,0**	21,4	22,3	20,9
Passive Aggressive Classifier	22,6	**23,8**	21,9	20,4	19,4
Perceptron	21,5	**23,8**	19,3	20,6	17,4
RandomForest	24,2	**24,6**	23,0	23,0	19,9
SGD	24,2	24,6	**25,0**	24,5	22,2
SVM	23,4	24,5	23,2	**25,6**	22,6

Tabela 10 – Taxa de acerto (%) dos modelos usando diferentes métodos de extração de características considerando a classificação com múltiplas classes. O melhor resultado por modelo está em negrito.

Conforme indicado em outros trabalhos, os resultados apresentados na Tabela 10 demonstram a eficiência das técnicas de extração de características TF-IDF e CountVectorizer no problema de detecção automática de *fake news*, sendo que em sua maioria, os classificadores apresentaram um desempenho superior quando usados em conjunto com estes dois extratores de características(VIJAYARAGHAVAN et al., 2020). Destaque para o classificador ExtraTrees que fazendo o uso do extrator de características CountVectorizer apresentou uma acurácia de 26,4%.

5.5.1 Oracle

Para verificar a viabilidade de aplicação de técnicas deMCS, semelhante ao discutido na Seção 5.4.2, foi analizando Oracle de cada conjunto composto por 5 versões de um mesmo classificador, cada versão fazendo uso de um dos extratores de características apresentados na Seção 5.2.1. Os resultados do Oracle, expostos na Tabela 11, demonstram que uma combinação ideal entre as máquinas de aprendizado testadas neste trabalho podem superar em larga margem os resultados da classificação monolítica apresentada na Tabela 10, indicando portanto a viabilidade do uso de técnicas de MCS.

Técnica	**Nº de Acertos**	**Taxa de acerto (%)**
AdaBoostClassifier	796	62,83
BernoulliNB	701	55,33
CNN	701	55,33
ComplementNB	671	52,96
ExtraTrees	753	59,43
GaussianNB	613	48,38
KNN	793	62,59
LSTM	793	62,59
LogisticRegression	712	56,20
MLP	754	59,51
MultinomialNB	525	41,44
Passive Aggressive Classifier	649	51,22
Perceptron	752	59,35
RandomForest	771	60,85
SGD	738	58,25
SVM	724	57,14
Todos	1067	84,21

Tabela 11 – Oracle por conjunto de classificadores fazendo uso de diferentes métodos de extração (CountVectorizer, TfIdf, Word2Vec, Glove e FastText) considerando a classificação com múltiplas classes.

5.5.2 Aplicação de técnicas de MCS

Semelhante ao apresentado na Seção 5.4.3, os resultados do Oracle, expostos na Tabela 11, indicam a possibilidade de que aplicação de técnicas de MCS para uma classificação envolvendo múltiplas classes podem apresentar melhorias no desempenho, se comparado com a classificação monolítica apresentada na Tabela 10. Portanto, foram aplicadas as técnicas apresentadas na Seção 3.4.1 nos mesmos conjuntos testados na Tabela 11. Os resultados são expostos na Tabela 12.

Técnica	Average	Max	Median	Min	Product	Voto Majoritário
AdaBoostClassifier	18,7	16,1	19,8	20,1	18,7	24,3
BernoulliNB	18,9	17,8	17,4	17,2	18,3	25,5
CNN	19,0	17,7	17,2	18,1	17,3	24,0
ComplementNB	17,0	16,7	17,2	17,0	17,1	26,0
ExtraTrees	17,8	17,4	17,7	18,4	18,7	24,0
GaussianNB	14,0	12,5	16,5	12,5	12,5	21,7
KNN	19,5	18,3	19,9	17,4	19,1	23,8
LSTM	19,3	18,7	20,5	18,4	19,9	21,9
LogisticRegression	17,3	17,6	18,0	16,9	17,4	24,9
MLP	15,6	15,6	16,2	14,8	15,9	24,2
MultinomialNB	19,3	19,7	20,2	19,4	19,5	20,9
Passive Aggressive Classifier	-	-	-	-	-	23,0
Perceptron	-	-	-	-	-	21,4
RandomForest	17,8	17,4	17,1	17,1	16,6	24,0
SGD	17,0	17,9	18,2	15,5	17,2	24,0
SVM	-	-	-	-	-	24,9
Todos	16,7	12,5	19,0	15,3	15,4	23,9

Tabela 12 – Acurácia por classificador considerando o problema com múltiplas classes. (%) Combinando um mesmo classificador fazendo uso dos extratores de características: CountVectorizer, TfIdf, Word2Vec, Glove e FastText.

É possível observar que, no caso da classificação com múltiplas classes, combinando os conjuntos de classificadores propostos na Tabela 11, não houve melhoria na classificação se comparado com a classificação monolítica.

5.5.3 Seleção de classificadores

Com o objetivo de selecionar os conjuntos de classificadores mais adequados para uma classificação conjunta, e que apresente melhoria em relação a classificação monolítica, apresentada na Tabela 10, foram aplicadas as técnicas discutidas no Capítulo 4, obtendo como resultado a Figura 4, que representa o comportamento de cada classificador em relação a todos os outros classificadores do *pool*.

5.5.3.1 Análise de comportamento

Para verificar a eficiência da técnica proposta neste trabalho, as 80 máquinas de aprendizado foram divididas em quatro grupos, cada um dos grupos compostos por 20 classificadores, separados de acordo com a indicação da distância nos seus comportamentos.

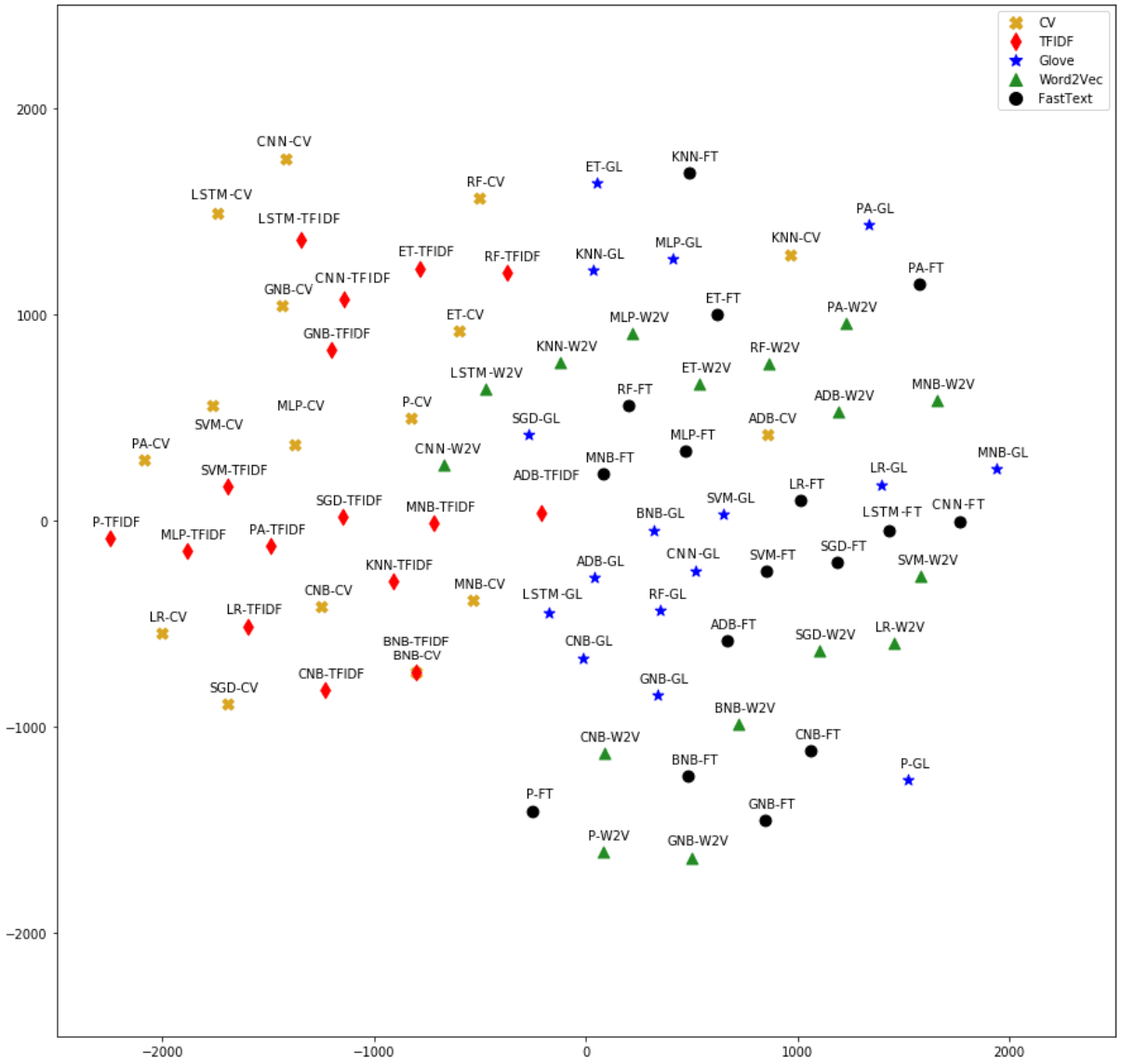


Figura 4 – t-SNE contendo todos os models utilizados neste trabalho considerando uma divisão com múltiplas classes

No Grupo 1, foram selecionados os 20 classificadores representados por pontos mais próximos das bordas na Figura 4. No Grupo 2, foram selecionados outros 20 classificadores, selecionando mais uma vez os classificadores mais próximos da borda excluindo os classificadores do Grupo 1. A mesma lógica é repetida para o Grupo 3, selecionando os 20 classificadores representados por pontos próximos a borda excluindo os classificadores do Grupo 1 e do Grupo 2. Os classificadores restantes compõem o Grupo 4, que são representados pelos pontos agrupados mais ao centro da Figura 4. A Figura 5 mostra como ficou esta divisão.

Analizando a Tabela 13, que apresenta o Oracle de cada um dos grupos, é possível observar uma queda gradual no Oracle de acordo com o afastamento dos pontos que representam os classificadores que integram estes grupos indicado na Figura 4.

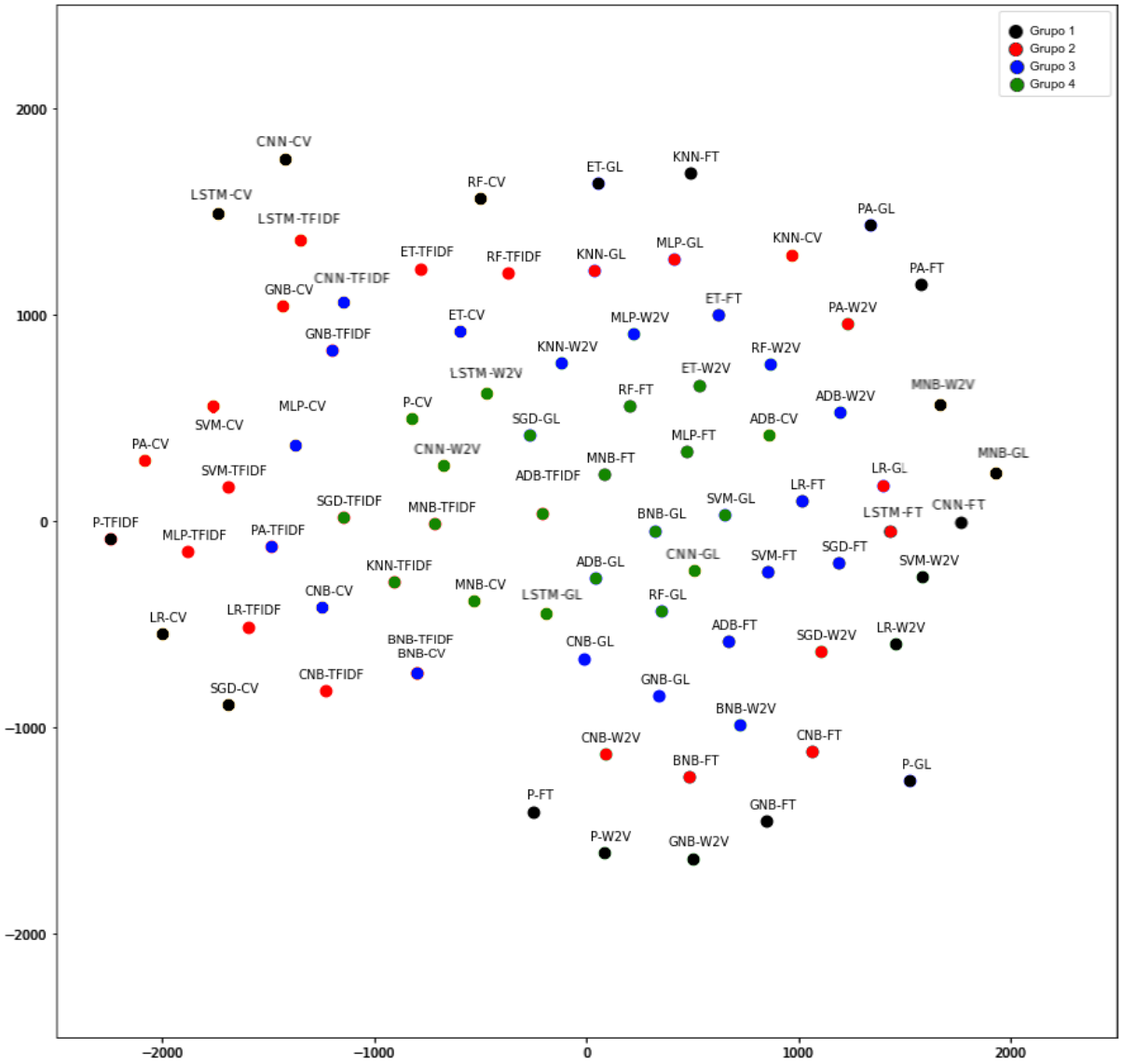


Figura 5 – Agrupamento de classificadores de acordo com a proximidade de comportamento

Técnica	**Nº de Acertos**	**Taxa de acerto (%)**
Grupo 1	1010	79,71
Grupo 2	940	74,19
Grupo 3	900	71,03
Grupo 4	880	69,45

Tabela 13 – Oracle por grupo de classificadores considerando a divisão apresentada na Figura 5

5.5.4 Resultados

Combinando os classificadores, P+TFIDF, SGD+FT, ET+CV, KNN+TFIDF, levando em conta a distância euclidiana entre a representação destes classificadores na Figura 4, e fazendo uso da técnica de combinação voto majoritário, apresentadas na Seção 3.4.1, foi obtido uma acurácia de 28,49%, desempenho superior a resultados apresentados na literatura (Tabela 14) que usam apenas o *statement* como valor de entrada e que tratam o LIAR dataset como um problema de múltiplas classes.

Método	Taxa de acerto (%)
Alhindi et al. (ALHINDI; PETRIDIS; MURESAN, 2018)	25,00
Wang. (WANG, 2017)	27,00
Proposta (4 classificadores + Voto majoritário)	28,49

Tabela 14 – Comparação dos resultados apresentados neste trabalho com artigos que realizaram experimentos similares

5.6 CLASSIFICAÇÃO FA-KES DATASET

Para garantir uma maior generalização da proposta desta dissertação, foram realizados experimentos usando um segundo conjunto de dados, o FA-KES Dataset. Seguindo outros trabalhos que fizeram uso deste dataset, os padrões foram divididos em duas partes, 80% foram usados no processo de treinamento dos classificadores, os 20% restantes foram destinados ao teste (ELHADAD; LI; GEBALI, 2019). Seguindo ainda o padrão de experimentação apresentados nas Seções 5.4 e 5.5, foi feita uma classificação monolítica do FA-KES com os 80 classificadores apresentados na Seção 5.2, os hiper-parâmetros utilizados em cada classificador, e o *range* utilizado pelo gridsearchcv no processo de busca da melhor configuração são apresentados na Tabela 21. O desempenho individual de cada classificador está exposto na Tabela 15, De maneira geral, os classificadores apresentaram um baixo desempenho quando testados individualmente, em alguns casos até mesmo inferior ao que se espera de uma classificação aleatória.

Técnica	**CV**	**TFIDF**	**W2V**	**FastText**	**GloVe**
AdaBoostClassifier	50,3	49,1	49,1	47,5	51,6
BernoulliNB	49,2	48,2	52,2	54,7	47,2
ComplementNB	52,8	53,4	51,6	48,4	54,0
CNN	50,3	54,9	46,7	49,9	49,9
ExtraTreesClassifier	49,7	47,8	50,9	52,2	52,2
GaussianNB	48,8	49,8	50,3	49,7	52,2
KNN	58,4	54,0	58,4	50,9	48,4
LSTM	47,5	51,4	52,2	52,0	53,1
LogisticRegression	52,8	53,4	52,8	52,2	54,7
MLPClassifier	50,3	50,9	52,2	53,4	50,9
MultinomialNB	53,4	51,6	52,2	53,4	57,8
PassiveAggressiveClassifier	49,4	52,2	47,8	47,8	47,8
Perceptron	48,8	49,7	52,2	52,2	52,2
RandomForestClassifier	49,1	56,5	52,8	55,9	46,6
SGDClassifier	51,6	51,6	52,8	49,8	52,8
SVM	51,6	51,6	52,8	47,8	52,8

Tabela 15 – Acurácia dos modelos usando diferentes métodos de extração de características considerando a classificação do FA-KES Dataset. O melhor resultado por modelo está em negrito.

5.6.1 Oracle

Observado o desempenho monolítico dos classificadores, o próximo passo para analisar a viabilidade da aplicação da proposta desta dissertação no conjunto de dados avaliado, e

verificar se uma eventual combinação entre os classificadores pode melhorar o desempenho em relação ao desempenho indivicual dos mesmos. Para tanto foi calculado o Oracle de 17 grupos de classificadores, sendo 16 deles formados por cinco versões de um mesmo classificador utilizando extratores de características diferentes, e um último grupo composto por todos os oitenta classificadores participantes do experimento. Os resultados são apresentados na Tabela 16, e indicam que há um cenário de diversidade no desempenho dos classificadores analisados, apontando variações nos padrões de acerto e erros dos mesmos, situação ideal para aplicação de técnicas de classificação combinadas.

Técnica	**Nº de Acertos**	**Taxa de acerto (%)**
AdaBoostClassifier	152	94,41
BernoulliNB	139	86,33
ComplementNB	147	91,3
CNN	126	78,86
ExtraTreesClassifier	138	85,71
GaussianNB	138	85,71
KNN	151	93,79
LSTM	146	91,20
LogisticRegression	136	84,47
MLPClassifier	137	85,09
MultinomialNB	129	80,12
PassiveAggressiveClassifier	122	75,78
Perceptron	131	81,37
RandomForestClassifier	138	85,71
SGDClassifier	160	99,38
SVM	145	90,06
Todos	161	100,0

Tabela 16 – Oracle por conjunto de classificadores fazendo uso de diferentes métodos de extração (CountVectorizer, TfIdf, Word2Vec, Glove e FastText) considerando a classificação com múltiplas classes.

5.6.2 Seleção de classificadores

Uma vez provado que existe a possibilidade de ganho de desempenho por meio do uso de combinação de classificadores, foi aplicada a proposta desta dissertação nos 80 classificadores experimentados. Para isso, 20% do conjunto de treinamento foi utilizado como validação. O resultado do comportamento de cada classificação analisado é apresentado na Figura 6, que apresenta uma visão panorâmica do comportamento de cada classificador em relação aos demais participantes do experimento.

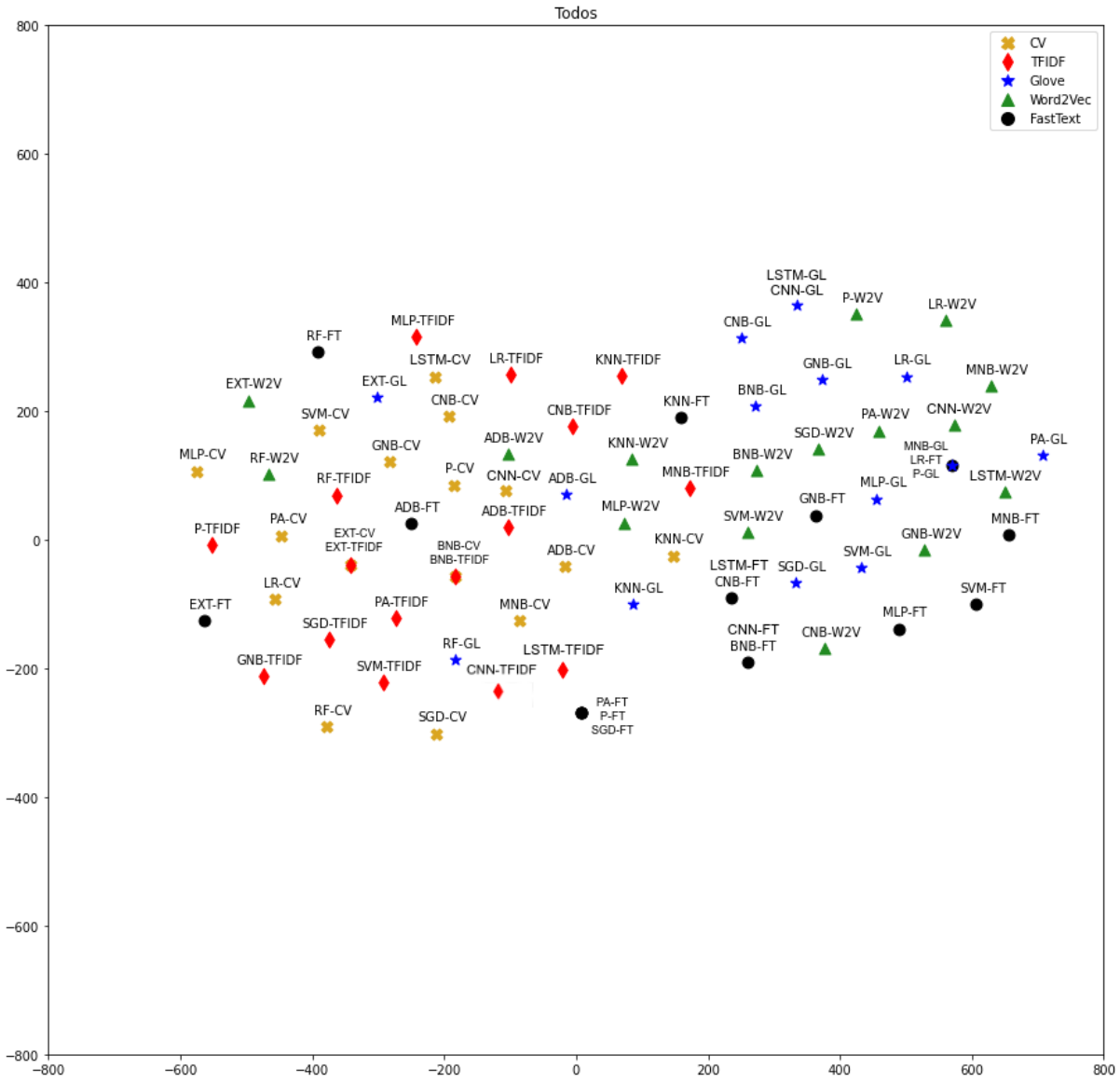


Figura 6 – Agrupamento de classificadores de acordo com a proximidade de comportamento

5.6.2.1 Análise de comportamento

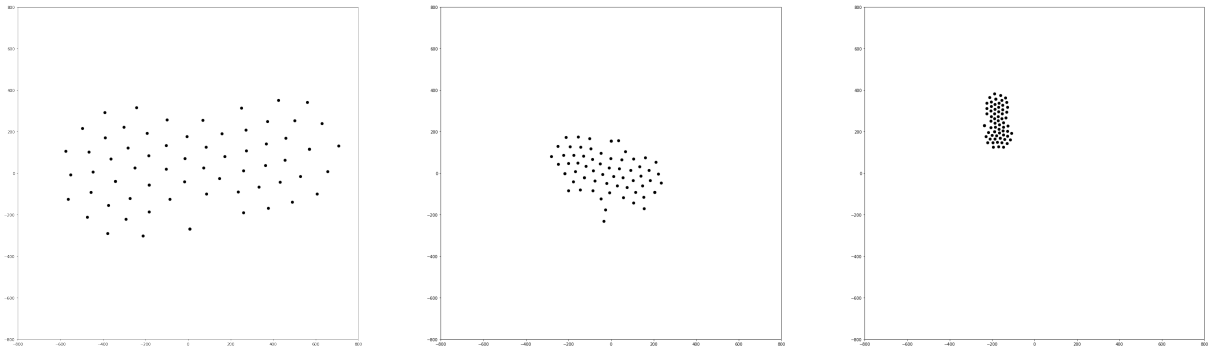
Conforme discutido no Capítulo 4, a aplicação da proposta apresentada neste trabalho resulta em um gráfico de comportamento dos classificadores, em que a proximidade na representação de dois classificadores indica semelhança no resultado de suas classificações. Seguindo ainda a explicação apresentada na Seção 3.1, o comportamento de um classificador esta diretamente relacionado com seu treinamento, sendo possível, piorar ou melhorar seu desempenho de acordo com a qualidade dos padrões usados neste processo.

Para assegurar que o comportamento de cada classificador em relação a todos os outros participantes do conjunto está bem representado na Figura 6, foi realizado um experimento dividido em três execuções, sendo que em cada execução, a qualidade dos padrões de treinamento sofreu uma redução, induzindo portanto, os classificadores a apresentar

comportamento com maior similaridade entre si. Em seguida, a diminuição na diversidade foi observada por meio da proposta desta dissertação.

Conforme apresentado na Seção 5.1.2, cada padrão contido no FA-KES Dataset conta com cinco campos, sendo que o campo *content* contribui com a maior quantidade de informações. Este campo conta com em média 2348 caracteres, e foi observado que a remoção deste campo no processo de treinamento piora o desempenho de todos os classificadores. Na primeira das três execuções do experimento, de forma idêntica ao processo que resultou na Figura 6, os classificadores foram treinados com o conjunto de dados completo, sem nenhuma alteração nos padrões de treinamento. Na segunda execução, com o intuito de induzir comportamento similar nos classificadores, o campo *content* foi limitado a 700 caracteres, desprezando qualquer informação que excedesse este limite. Já na terceira execução, os classificadores foram treinados somente com o *title* e o *content* de cada padrão, sendo que este último foi limitado a 380 caracteres. Vale ressaltar que o número de padrões para o treinamento dos classificadores se manteve o mesmo em todas as execuções, a limitação ocorreu apenas na qualidade cada padrão, restringindo a quantidade de informações disponíveis para os classificadores.

Como esperado, a piora na qualidade dos padrões de treinamento se refletiu no Oracle, sendo que, na primeira execução, o Oracle apresentado pelo conjunto de classificadores foi de 100,00%, e os classificadores apresentaram uma média de acerto de 51,13%, na segunda estes valores caíram, sendo que o Oracle apresentado ficou em 96% e o desempenho médio dos classificadores ficou em 49,77%, e na terceira e última execução o conjunto apresentou um Oracle de 84,00% e média individual de acerto por classificador de 48,20%.



(a) Execução 1 (b) Execução 2 (c) Execução 3

Figura 7 – Análise do comportamento dos classificadores a medida que os mesmos são induzidos a apresentar comportamento mais semelhante entre sí.

A limitação na entrada de dados para o treinamento dos classificadores implicou em uma queda na diversidade do conjunto, fato que pode ser comprovado comparando as reduções no desempenho médio da classificação individual em contraste com a redução sofrida no Oracle. Entre a primeira e a terceira execução o desempenho da classificação individual caiu apenas 2,93%, já o Oracle apresentou redução de 16,00%. Este é um

indicativo de que o experimento induziu os classificadores a apresentar comportamento similar. Individualmente, não houve grande mudança nos resultados da classificação, mas houve uma redução significativa no desempenho do conjunto. Sabendo disso, foi aplicada a proposta deste trabalho no fim de cada execuções, a Figura 7 é o resultado deste processo. Considerando que foram mantidas as mesmas dimensões entre os gráficos, a queda na diversidade dos conjuntos se refletiu no resultado do processo, que apresenta pontos mais agrupados de acordo com a redução dos dados disponíveis para treinamento, e portanto, a redução na diversidade. Este comportamento está coerente com o objetivo da proposta, indicando que a Figura 6 é fiel em representar a diversidade dos classificadores analisados.

Este experimento demonstra também a dificuldade de compor um *pool* adequado para classificação combinada, sendo impossível descobrir se um classificador contribui ou não para o conjunto analisando apenas o seu resultado individual.

5.6.3 Aplicação de técnicas de MCS

Para avaliar o desempenho de regras fixas de combinação de classificadores no dataset proposto, foram aplicadas as técnicas: *Average, Max, Median, Min, Product* e *Voto Majoritário* nos conjuntos avaliados na Tabela 16. Os resultados são apresentados na Tabela 17. Comparando os resultados das classificações monolíticas, apresentados na Tabela 15, o Oracle dos conjuntos de classificadores apresentado na Tabela 16 e a aplicação de regras fixas de combinações nos mesmos conjuntos em que se avaliou o Oracle, Tabela 17, pode-se concluir que uma combinação de classificadores pode melhorar em grande medida o desempenho final da classificação, no entanto, deve-se encontrar o *pool* ideal para esta combinação, e eventualmente, avaliar técnicas de combinação que explore de maneira adequada este *pool*.

5.6.4 Resultados

Fazendo uso da vantagem resultante de aplicação da proposta desta dissertação, foram experimentadas diversas configurações de *pool* com regras fixas de combinação, sendo que o melhor resultado foi obtido combinando os classificadores: RF+CV, RF+TFIDF, RF+FT e MNB+CV, usando a técnica de combinação Voto Majoritário alcançando uma taxa de acerto de 59,60%. A Tabela 18 faz uma comparação entre o resultado obtido neste trabalho e uma publicação que realiza experimento com o mesmo conjunto de dados.

Técnica	Average	Max	Median	Min	Product	Voto Majoritário
AdaBoostClassifier	51,7	50,1	49,3	50,1	49,1	50,3
BernoulliNB	51,9	44,1	50,3	50,1	43,3	56,6
CNN	48,0	48,7	48,9	49,2	50,0	50,0
ComplementNB	51,2	50,4	50,2	47,2	49,9	52,2
ExtraTrees	50,0	49,8	51,3	44,6	47,7	50,3
GaussianNB	48,1	42,5	47,2	50,1	50,5	47,9
KNN	49,1	50,3	50,9	51,2	49,2	41,8
LSTM	49,3	48,8	44,3	44,4	50,9	51,9
LogisticRegression	49,7	50,6	50,0	44,9	49,3	50,9
MLP	51,6	49,4	46,1	49,3	44,8	40,3
MultinomialNB	49,3	50,3	44,4	45,4	45,5	44,9
Passive Aggressive	-	-	-	-	-	51,0
Perceptron	-	-	-	-	-	52,4
RandomForest	50,9	51,4	48,3	50,3	50,1	47,3
SGD	51,1	50,1	50,0	48,5	42,2	46,6
SVM	-	-	-	-	-	24,9
Todos	46,2	50,5	49,4	50,8	40,4	41,9

Tabela 17 – Acurácia por técnica de combinação de um mesmo classificador fazendo uso dos extratores de características: CountVectorizer, TfIdf, Word2Vec, Glove e FastText.

Método	Taxa de acerto (%)
Elhadad, Mohamed K. et al. (ELHADAD; LI; GEBALI, 2019)	58.09
Proposta (4 classificadores + Voto)	59,60

Tabela 18 – Comparação dos resultados apresentados neste trabalho com artigos que realizaram experimentos similares

6 CONCLUSÃO

Este capítulo apresenta as considerações finais sobre o trabalho desenvolvido nesta dissertação. A Seção 6.1 traz considerações sobre os resultados observados. Na Seção 6.2 são destacadas as principais contribuições desta dissertação e na Seção 6.3 algumas sugestões de trabalhos futuros são discutidas.

6.1 CONSIDERAÇÕES FINAIS

A implementação de um sistema de *automated fact-checking* é uma tarefa de difícil execução. Tendo em vista que uma declaração falsa é fabricada com o objetivo de enganar o receptor, selecionando fatos verídicos e os mesclando com informações falsas, ou muitas vezes, omitindo parte da informação, o que posiciona tal declaração em um ponto nebuloso entre verdadeira e falsa, soma-se a esta dificuldade, a complexidade de selecionar um grupo de classificadores para aplicação de técnicas de MCS. Os experimentos realizados nas Seções 5.5.3.1, 5.4.2.1 e em especial na Seção 5.6.2.1, deixam clara a dificuldade desta tarefa, sendo impossível, analisando apenas o resultado de um classificador, saber se o mesmo vai contribuir com a composição do *pool*. Outra evidência desta complexidade pode ser observada analisando o desempenho individual de diversas técnicas de classificação, que muitas vezes, têm o seus resultados próximo do que se espera de uma classificação aleatória.

Sabendo que técnicas de combinações de classificadores têm se mostrado eficientes na melhoria do desempenho de diversos problemas de *machine learning*, e que, para se obter uma classificação em conjunto que supere uma classificação monolítica é importante que os membros deste conjunto apresentem comportamento heterogêneo entre si. Este trabalho apresentou uma proposta em que, dado um *pool* de classificadores, é possível observar de maneira visual a proximidade do comportamento de cada classificador, possibilitando portanto, selecionar um subgrupo de classificadores que apresente comportamento mais heterogêneo entre seus membros.

A metodologia proposta foi aplicada em dois conjuntos de dados já testado em outros trabalhos. Inicialmente foi considerada uma classificação binária do Liar Dataset, e em seguida, considerando o problema de classificação com múltiplas classes para o mesmo banco, e por fim, foram realizados experimentos com o FA-KES Dataset. A proposta se mostrou eficiente nos todos casos, apresentando uma visão panorâmica do comportamento dos classificadores, tornando possível identificar quais classificadores dentro do *pool* têm comportamento semelhante e quais apresentam comportamento distinto. Em seguida, sabendo identificar quais classificadores apresentam melhor possibilidade de ganho de desempenho em uma classificação conjunta, foram experimentadas técnicas de combinação

nos subgrupos selecionados. Em todas as situações, classificação binária e classificação com múltiplas classes, a combinação dos subgrupos apresentou melhora de desempenho se comparado a classificação monolítica do *pool*, e, se comparada a outros trabalhos publicados que fizeram uso do mesmo conjunto de dados, comprovando portanto, a eficiência da proposta desta dissertação.

6.2 CONTRIBUIÇÕES

Destacam-se como contribuições deste trabalho a comprovação, por meio do Oracle calculado em diversas grupos de classificadores, de que uma combinação de classificadores pode elevar em grande escala o *state of the art* do problema de detecção automática de *fake news*, podendo atingir uma acurácia de 100% em uma classificação binária, e 84,21% em uma classificação com múltiplas classes no conjunto de dados analisado.

Vale citar também, os resultados apresentados neste trabalho que superam os resultados de publicações que realizaram experimentos comparáveis. Usando a técnica de combinação *Median* no grupo formado por cinco versões do classificador SGD, cada versão fazendo uso de um dos extratores de características utilizados neste trabalho, conseguiu-se uma acurácia de 63,10% na classificação binária do conjunto de dados. No entanto, o melhor resultado alcançado na classificação binária, foi obtido pelos classificadores: MultinomialNB + Glove, SGD + CountVectorizer, RandomForest + Word2Vec, KNN + Word2Vec, grupo que, quando combinado pela regra *Average* apresentou uma taxa de acerto de 64,70%. Na classificação considerando múltiplas classes, o melhor resultado foi apresentado pelos classificadores: Perceptron+TFIDF, SGD+FastText, ExtraTrees+Countvectorizer e KNN+TFIDF, que quando combinados pela regra de voto majoritário, apresentaram uma acurácia de 28,49%. Ressalta-se que a composição dos dois grupos que apresentaram melhores resultados, só foi possível devido à aplicação da proposta apresentada neste trabalho, sendo que estes resultados são considerados ao mesmo tempo, uma contribuição e uma validação da metodologia proposta.

A principal contribuição, no entanto, é a proposta da metodologia, que dado um grupo de classificadores, apresenta uma técnica de seleção dos subconjuntos mais promissores para uma classificação combinada.

6.3 TRABALHOS FUTUROS

Como sugestão de continuidade para este trabalho, está à aplicação da metodologia aqui descrita e experimentada a novos conjuntos de dados, novos algoritmos classificadores e novos extratores de características. Fica também como sugestão, uma análise do impacto da adição dos demais atributos do conjunto de dados experimentados neste trabalho no resultado da classificação final.

Entendendo que uma combinação de classificadores bem sucedida é composta pela seleção de um grupo adequado de classificadores para compor um *pool*, em conjunto com uma função que consiga tirar proveito da diversidade oferecida por este *pool*, e que este trabalho apresenta uma forma eficiente de encontrar este grupo, fica como sugestão para trabalhos futuro o uso da metodologia proposta neste trabalho em conjunto com técnicas de seleção dinâmica de classificadores (CRUZ et al., 2020).

Considerando a natureza do problema de detecção de *fake news*, uma outra sugestão é o uso conjunto da proposta apresentada nesta dissertação e técnicas de análise de sentimentos.

Por fim,entendendo que o idioma da notícia é fator delicado no processo de extração de características, e que pode apresentar variações nos resultados apresentados, tendo em vista que a proposta desta dissertação foi aplicada somente em conjuntos de dados com língua inglesa, se sugere uma análise do comportamento da metodologia proposta neste trabalho em um conjunto de dados em língua portuguesa (MORENO; BRESSAN, 2019).

REFERÊNCIAS

ALHINDI, T.; PETRIDIS, S.; MURESAN, S. Where is your evidence: Improving fact-checking by justification modeling. In: *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*. Brussels, Belgium: Association for Computational Linguistics, 2018. p. 85–90.

ALLCOTT, H.; GENTZKOW, M. Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, v. 31, n. 2, p. 211–36, May 2017.

ARNAUDO, D. Computational propaganda in brazil: social bots during elections. Computational Propaganda Research Project, v. 2017.8, n. 2017, p. 1–39, 2017.

BAKIR, V.; MCSTAY, A. Fake news and the economy of emotions. *Digital Journalism*, Routledge, v. 6, n. 2, p. 154–175, 2018.

Bhutani, B.; Rastogi, N.; Sehgal, P.; Purwar, A. Fake news detection using sentiment analysis. In: *2019 Twelfth International Conference on Contemporary Computing (IC3)*. [S.l.: s.n.], 2019. p. 1–5.

BREIMAN, L. Bagging predictors. *Machine Learning*, v. 24, n. 2, p. 123–140, ago. 1996. ISSN 1573-0565. Disponível em: <https://doi.org/10.1007/BF00058655>.

BREIMAN, L. Random forests. *Machine Learning*, Kluwer Academic Publishers, v. 45, n. 1, p. 5–32, 2001. ISSN 0885-6125. Disponível em: <http://dx.doi.org/10.1023/A%3A1010933404324>.

BROWN, G.; KUNCHEVA, L. I. “good” and “bad” diversity in majority vote ensembles. In: GAYAR, N. E.; KITTLER, J.; ROLI, F. (Ed.). *Multiple Classifier Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 124–133. ISBN 978-3-642-12127-2.

CAVALCANTI, G. D.; OLIVEIRA, L. S.; MOURA, T. J.; CARVALHO, G. V. Combining diversity measures for ensemble pruning. *Pattern Recognition Letters*, Elsevier Science Inc., New York, NY, USA, v. 74, n. C, p. 38–45, abr. 2016. ISSN 0167-8655.

CINELLI, M.; QUATTROCIOCCHI, W.; GALEAZZI, A.; VALENSISE, C. M.; BRUGNOLI, E.; SCHMIDT, A. L.; ZOLA, P.; ZOLLO, F.; SCALA, A. The covid-19 social media infodemic. *CoRR*, abs/2003.05004, 2020.

Coelho, G. P.; Von Zuben, F. J. The influence of the pool of candidates on the performance of selection and combination techniques in ensembles. In: *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. [S.l.: s.n.], 2006. p. 5132–5139.

CONROY, N.; RUBIN, V.; CHEN, Y. Automatic deception detection: Methods for finding fake news. *Proceedings of the Association for Information Science and Technology*, v. 52, p. 1–4, 01 2015.

COX, M. A. A.; COX, T. F. Multidimensional scaling. In: _____. *Handbook of Data Visualization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 315–347. ISBN 978-3-540-33037-0. Disponível em: <https://doi.org/10.1007/978-3-540-33037-0_14>.

CRUZ, R. M. O.; CAVALCANTI, G. D. C.; TSANG, I. R.; SABOURIN, R. Feature representation selection based on classifier projection space and oracle analysis. *Expert Syst. Appl.*, v. 40, n. 9, p. 3813–3827, 2013.

CRUZ, R. M. O.; HAFEMANN, L. G.; SABOURIN, R.; CAVALCANTI, G. D. C. Deslib: A dynamic ensemble selection library in python. *Journal of Machine Learning Research*, v. 21, n. 8, p. 1–5, 2020.

CRUZ, R. M. Oliveira e. *Methods for dynamic selection and fusion of ensemble of classifiers*. 290 p. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2011.

CUI, L.; LEE, D. *CoAID: COVID-19 Healthcare Misinformation Dataset*. 2020.

Das Bhattacharjee, S.; Talukder, A.; Balantrapu, B. V. Active learning based news veracity detection with feature weighting and deep-shallow fusion. In: *2017 IEEE International Conference on Big Data (Big Data)*. [S.l.: s.n.], 2017. p. 556–565.

Das, S.; Ghosh, S.; Bhattacharya, S.; Varma, R.; Bhandari, D. Critical dimension of word2vec. In: *2019 2nd International Conference on Innovations in Electronics, Signal Processing and Communication (IESC)*. [S.l.: s.n.], 2019. p. 202–206.

DEBNATH, A.; RAHMAN, R.; ISLAM, M. M.; RAZZAQUE, M. A. A hierarchical learning model for claim validation. In: UDDIN, M. S.; BANSAL, J. C. (Ed.). *Proceedings of International Joint Conference on Computational Intelligence*. Singapore: Springer Singapore, 2020. p. 431–441. ISBN 978-981-13-7564-4.

DIETTERICH, T. G. Ensemble methods in machine learning. In: *Multiple Classifier Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000. p. 1–15. ISBN 978-3-540-45014-6.

DROZD, A.; GLADKOVA, A.; MATSUOKA, S. Word embeddings, analogies, and machine learning: Beyond king - man + woman = queen. In: CALZOLARI, N.; MATSUMOTO, Y.; PRASAD, R. (Ed.). *COLING*. [S.l.]: ACL, 2016. p. 3519–3530. ISBN 978-4-87974-702-0.

ELHADAD, M. K.; LI, K. F.; GEBALI, F. A novel approach for selecting hybrid features from online news textual metadata for fake news detection. In: BAROLLI, L.; HELLINCKX, P.; NATWICHAI, J. (Ed.). *3PGCIC*. Springer, 2019. (Lecture Notes in Networks and Systems, v. 96), p. 914–925. ISBN 978-3-030-33509-0. Disponível em: <http://dblp.uni-trier.de/db/conf/3pgcic/3pgcic2019.html#ElhadadLG19>.

ESPADOTO, M.; RODRIGUES, F. C. M.; HIRATA, N. S. T.; JR., R. H.; TELEA, A. C. Deep learning inverse multidimensional projections. In: LANDESBERGER, T. von; TURKAY, C. (Ed.). *EuroVA@EuroVis*. Eurographics Association, 2019. p. 13–17. ISBN 978-3-03868-087-1. Disponível em: <http://dblp.uni-trier.de/db/conf/vissym/va2019.html#EspadotoRHHT19>.

FREUND, Y.; SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, v. 55, n. 1, p. 119 – 139, 1997. ISSN 0022-0000. Disponível em: <http://www.sciencedirect.com/science/article/pii/S002200009791504X>.

Fumera, G.; Roli, F. A theoretical and experimental analysis of linear combiners for multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 27, n. 6, p. 942–956, June 2005.

GAMA, J.; SEBASTIãO, R.; RODRIGUES, P. P. On evaluating stream learning algorithms. *Mach. Learn.*, v. 90, n. 3, p. 317–346, 2013. Disponível em: <http://dblp.uni-trier.de/db/journals/ml/ml90.html#GamaSR13>.

GERS, F. Learning to forget: continual prediction with lstm. *IET Conference Proceedings*, Institution of Engineering and Technology, p. 850–855(5), January 1999. Disponível em: <https://digital-library.theiet.org/content/conferences/10.1049/cp_19991218>.

GEURTS, P.; ERNST, D.; WEHENKEL, L. Extremely randomized trees. *Mach. Learn.*, v. 63, n. 1, p. 3–42, 2006. Disponível em: <http://dblp.uni-trier.de/db/journals/ml/ml63.html#GeurtsEW06>.

GIACINTO, G.; ROLI, F. Design of effective neural network ensembles for image classification purposes. *Image Vision Comput.*, v. 19, n. 9-10, p. 699–707, 2001.

GOUTTE, C.; GAUSSIER, E. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In: LOSADA, D. E.; FERNÁNDEZ-LUNA, J. M. (Ed.). *Advances in Information Retrieval*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. p. 345–359. ISBN 978-3-540-31865-1.

GRAVES, D. Understanding the promise and limits of automated fact-checking. *Technical report, Reuters Institute, University of Oxford*, 2018.

GUO, C.; CAO, J.; ZHANG, X.; SHU, K.; YU, M. Exploiting emotions for fake news detection on social media. *CoRR*, abs/1903.01728, 2019. Disponível em: <http://dblp.uni-trier.de/db/journals/corr/corr1903.html#abs-1903-01728>.

HANSELOWSKI, A.; S., A. P. V.; SCHILLER, B.; CASPELHERR, F.; CHAUDHURI, D.; MEYER, C. M.; GUREVYCH, I. A retrospective analysis of the fake news challenge stance detection task. *CoRR*, abs/1806.05180, 2018. Disponível em: <http://arxiv.org/abs/1806.05180>.

HASSAN, N.; ADAIR, B.; HAMILTON, J.; LI, C.; TREMAYNE, M.; YANG, J.; YU, C. The quest to automate fact-checking. *Proceedings of the 2015 Computation + Journalism Symposium*, 10 2015.

HO, T. K. Multiple classifier combination: Lessons and next steps. In: _____. [S.l.]: World Scientific, 2001. (Series in Machine Perception and Artificial Intelligence, v. 47), cap. 7, p. 171–198.

HORNE, B. D.; ADALI, S. This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. *CoRR*, abs/1703.09398, 2017. Disponível em: <http://dblp.uni-trier.de/db/journals/corr/corr1703.html#HorneA17>.

Isaak, J.; Hanna, M. J. User data privacy: Facebook, cambridge analytica, and privacy protection. *Computer*, v. 51, n. 8, p. 56–59, August 2018. ISSN 1558-0814.

JOULIN, A.; GRAVE, E.; BOJANOWSKI, P.; DOUZE, M.; JéGOU, H.; MIKOLOV, T. Fasttext.zip: Compressing text classification models. *CoRR*, abs/1612.03651, 2016.

JOULIN, A.; GRAVE, E.; BOJANOWSKI, P.; MIKOLOV, T. Bag of tricks for efficient text classification. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, 2017. p. 427–431.

JR., E. C. T.; LIM, Z. W.; LING, R. Defining "fake news". *Digital Journalism*, Routledge, v. 6, n. 2, p. 137–153, ago. 2018.

KALCHBRENNER, N.; GREFENSTETTE, E.; BLUNSOM, P. *A Convolutional Neural Network for Modelling Sentences*. 2014. Cite arxiv:1404.2188. Disponível em: <http://arxiv.org/abs/1404.2188>.

KARIMI, H.; ROY, P.; SABA-SADIYA, S.; TANG, J. Multi-source multi-class fake news detection. In: BENDER, E. M.; DERCZYNSKI, L.; ISABELLE, P. (Ed.). *COLING*. [S.l.]: Association for Computational Linguistics, 2018. p. 1546–1557. ISBN 978-1-948087-50-6.

KHALDAROVA, I.; PANTTI, M. Fake news. *Journalism Practice*, Routledge, v. 10, n. 7, p. 891–901, 2016.

KHAN, J. Y.; KHONDAKER, M. T. I.; IQBAL, A.; AFROZ, S. A benchmark study on machine learning methods for fake news detection. *CoRR*, abs/1905.04749, 2019.

KIRILIN, A.; STRUBE, M. Exploiting a speaker's credibility to detect fake news. In: *Proceedings of Data Science, Journalism & Media workshop at KDD (DSJM'18)*. [S.l.: s.n.], 2018.

Kittler, J.; Hatef, M.; Duin, R. P. W.; Matas, J. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 20, n. 3, p. 226–239, March 1998. ISSN 0162-8828.

KOTSIANTIS, S. B. Supervised machine learning: A review of classification techniques. *Informatica*, v. 31, n. 3, p. 249–268, 2007. Disponível em: <http://www.informatica.si/PDF/31-3/11_Kotsiantis%20-%20Supervised%20Machine%20Learning%20-%20A%20Review%20of...pdf>.

KRUIGER, J. F.; RAUBER, P. E.; MARTINS, R. M.; KERREN, A.; KOBOUROV, S. G.; TELEA, A. Graph layouts by t-sne. *Comput. Graph. Forum*, v. 36, n. 3, p. 283–294, 2017.

KUCHARSKI, A. Post-truth: Study epidemiology of fake news. *Nature*, v. 540, p. 525–525, 12 2016.

KULKARNI, A.; SHIVANANDA, A. Advanced natural language processing. In: _____. *Natural Language Processing Recipes: Unlocking Text Data with Machine Learning and Deep Learning using Python*. Berkeley, CA: Apress, 2019. p. 97–128. ISBN 978-1-4842-4267-4.

Kuncheva, L. I. A theoretical study on six classifier fusion strategies. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 24, n. 2, p. 281–286, Feb 2002. ISSN 0162-8828.

Kuncheva, L. I.; Whitaker, C. J. Ten measures of diversity in classifier ensembles: limits for two classifiers. In: *A DERA/IEE Workshop on Intelligent Sensor Processing (Ref. No. 2001/050)*. [S.l.: s.n.], 2001. p. 10/1–1010.

KUNCHEVA, L. I.; WHITAKER, C. J. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Mach. Learn.*, v. 51, n. 2, p. 181–207, 2003.

LAI, C.-C.; SHIH, T.-P.; KO, W.-C.; TANG, H.-J.; HSUEH, P.-R. Severe acute respiratory syndrome coronavirus 2 (sars-cov-2) and coronavirus disease-2019 (covid-19): The epidemic and the challenges. *International Journal of Antimicrobial Agents*, v. 55, n. 3, p. 105924, 2020. ISSN 0924-8579.

LANGLEY, P.; SIMON, H. A. Applications of machine learning and rule induction. *Commun. ACM*, v. 38, n. 11, p. 54–64, 1995.

LAZER, D. M. J.; BAUM, M. A.; BENKLER, Y.; BERINSKY, A. J.; GREENHILL, K. M.; MENCZER, F.; METZGER, M. J.; NYHAN, B.; PENNYCOOK, G.; ROTHSCHILD, D.; SCHUDSON, M.; SLOMAN, S. A.; SUNSTEIN, C. R.; THORSON, E. A.; WATTS, D. J.; ZITTRAIN, J. L. The science of fake news. *Science*, American Association for the Advancement of Science, v. 359, n. 6380, p. 1094–1096, March 2018. ISSN 0036-8075.

LEBANON, G.; MAO, Y.; DILLON, J. V. The locally weighted bag of words framework for document representation. *J. Mach. Learn. Res.*, v. 8, p. 2405–2441, 2007.

LEVY, O.; GOLDBERG, Y. Neural word embedding as implicit matrix factorization. In: GHAHRAMANI, Z.; WELLING, M.; CORTES, C.; LAWRENCE, N. D.; WEINBERGER, K. Q. (Ed.). *NIPS*. [S.l.: s.n.], 2014. p. 2177–2185.

MAATEN, L. van der; HINTON, G. Visualizing data using t-SNE. *Journal of Machine Learning Research*, v. 9, p. 2579–2605, 2008.

MANNING, C. D.; SCHÜTZE, H. *Foundations of statistical natural language processing*. Cambridge, Mass.: MIT Press, 1999. ISBN 0262133601 9780262133609.

MARQUéS, A. I.; GARCíA, V.; SáNCHEZ, J. S. Exploring the behaviour of base classifiers in credit scoring ensembles. *Expert Syst. Appl.*, v. 39, n. 11, p. 10244–10250, 2012. Disponível em: <http://dblp.uni-trier.de/db/journals/eswa/eswa39.html#MarquesGS12>.

MCINNES, L.; HEALY, J.; MELVILLE, J. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. 2018. Cite arxiv:1802.03426Comment: Reference implementation available at http://github.com/lmcinnes/umap. Disponível em: <http://arxiv.org/abs/1802.03426>.

MIKOLOV, T.; GRAVE, E.; BOJANOWSKI, P.; PUHRSCH, C.; JOULIN, A. Advances in pre-training distributed word representations. In: *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*. [S.l.: s.n.], 2018.

MORENO, J. a.; BRESSAN, G. Factck.br: A new dataset to study fake news. In: *Proceedings of the 25th Brazillian Symposium on Multimedia and the Web*. New York, NY, USA: Association for Computing Machinery, 2019. (WebMedia '19), p. 525–527. ISBN 9781450367639.

MORGAN, S. Fake news, disinformation, manipulation and online tactics to undermine democracy. *Journal of Cyber Policy*, Routledge, v. 3, n. 1, p. 39–43, 2018.

PAL, N. R.; PAL, S. Editorial: Computational intelligence for pattern recognition. *IJPRAI*, v. 16, n. 7, p. 773–780, 2002. Disponível em: <http://dblp.uni-trier.de/db/journals/ijprai/ijprai16.html#PalP02>.

PEKALSKA, E.; DUIN, R.; SKURICHINA, M. A discussion on the classifier projection space for classifier combining. In: . [S.l.: s.n.], 2002. v. 2364, p. 137–148.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. v. 14, p. 1532–1543, 2014.

PERSILY, N. The 2016 u.s. election: Can democracy survive the internet? *Journal of Democracy*, v. 28, n. 2, p. 63–76, 2017.

Ponti Jr., M. P. Combining classifiers: From the creation of ensembles to the decision fusion. In: *2011 24th SIBGRAPI Conference on Graphics, Patterns, and Images Tutorials*. [S.l.: s.n.], 2011. p. 1–10.

RASCHKA, S.; MIRJALILI, V. *Python Machine Learning: Machine Learning and Deep Learning with Python, Scikit-Learn, and TensorFlow 2, 3rd Edition*. Packt Publishing, 2019. (Expert insight). ISBN 9781789955750. Disponível em: <https://books.google.com.br/books?id=n1cjyAEACAAJ>.

RASHKIN, H.; CHOI, E.; JANG, J. Y.; VOLKOVA, S.; CHOI, Y. Truth of varying shades: Analyzing language in fake news and political fact-checking. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, 2017. p. 2931–2937.

Reis, J. C. S.; Correia, A.; Murai, F.; Veloso, A.; Benevenuto, F. Supervised learning for fake news detection. *IEEE Intelligent Systems*, v. 34, n. 2, p. 76–81, 2019.

RESENDE, G.; MESSIAS, J.; SILVA, M.; ALMEIDA, J. M.; VASCONCELOS, M.; BENEVENUTO, F. A system for monitoring public political groups in whatsapp. In: NETO, M. C. M.; NOVAIS, R. L.; FERRAZ, C.; VIANA, W. (Ed.). *WebMedia*. ACM, 2018. p. 387–390. ISBN 978-1-4503-5867-5. Disponível em: <http://dblp.uni-trier.de/db/conf/webmedia/webmedia2018.html#ResendeMSAVB18>.

RIBEIRO, F. e. a. Media bias monitor: Quantifying biases of social media outlets at large-scale. 2018.

RIEDEL, B.; AUGENSTEIN, I.; SPITHOURAKIS, G. P.; RIEDEL, S. A simple but tough-to-beat baseline for the fake news challenge stance detection task. *CoRR*, abs/1707.03264, 2017. Disponível em: <http://arxiv.org/abs/1707.03264>.

ROLI, F. Multiple classifier systems. In: _____. *Encyclopedia of Biometrics*. Boston, MA: Springer US, 2009. p. 981–986. ISBN 978-0-387-73003-5.

ROLI, F.; KITTLER, J. (Ed.). *Multiple Classifier Systems, Third International Workshop, MCS 2002, Cagliari, Italy, June 24-26, 2002, Proceedings*, v. 2364 de *Lecture Notes in Computer Science*, (Lecture Notes in Computer Science, v. 2364). Springer, 2002. ISBN 3-540-43818-1. Disponível em: <http://dblp.uni-trier.de/db/conf/mcs/mcs2002.html>.

RONG, X. word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*, 2014.

ROSE, J. Brexit, trump, and post-truth politics. *Public Integrity*, Routledge, v. 19, n. 6, p. 555–558, 2017.

SALEM, F. K. A.; FEEL, R. A.; ELBASSUONI, S.; JABER, M.; FARAH, M. Fa-kes: A fake news dataset around the syrian war. In: PFEFFER, J.; BUDAK, C.; LIN, Y.-R.; MORSTATTER, F. (Ed.). *ICWSM*. AAAI Press, 2019. p. 573–582. Disponível em: <http://dblp.uni-trier.de/db/conf/icwsm/icwsm2019.html#SalemFEJF19>.

SALTON, G.; BUCKLEY, C. Term-Weighting Approaches In Automatic Text Retrieval. *Information Processing and Management*, v. 24, p. 513–523, 1988.

Sarakit, P.; Theeramunkong, T.; Haruechaiyasak, C.; Okumura, M. Classifying emotion in thai youtube comments. In: *2015 6th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES)*. [S.l.: s.n.], 2015. p. 1–5.

SCHAPIRE, R. E. The Strength of Weak Learnability. *Machine Learning*, v. 5, p. 197–227, 1990.

SHABAN, T. A.; HEXTER, L.; CHOI, J. D. Event analysis on the 2016 u.s. presidential election using social media. In: CIAMPAGLIA, G. L.; MASHHADI, A. J.; YASSERI, T. (Ed.). *SocInfo (1)*. [S.l.]: Springer, 2017. (Lecture Notes in Computer Science, v. 10539), p. 201–217. ISBN 978-3-319-67217-5.

SHAO, C.; CIAMPAGLIA, G. L.; VAROL, O.; FLAMMINI, A.; MENCZER, F. The spread of fake news by social bots. *CoRR*, abs/1707.07592, 2017.

SKALAK, D. B. The sources of increased accuracy for two proposed boosting algorithms. In: *In Proc. American Association for Arti Intelligence, AAAI-96, Integrating Multiple Learned Models Workshop*. [S.l.: s.n.], 1996. p. 120–125.

SPOHR, D. Fake news and ideological polarization: Filter bubbles and selective exposure on social media. *Business Information Review*, v. 34, n. 3, p. 150–160, 2017.

SULLIVAN, M. C. Why librarians can't fight fake news. *Journal of Librarianship and Information Science*, v. 51, n. 4, p. 1146–1156, 2019.

SYARIF, I.; ZALUSKA, E.; PRUGEL-BENNETT, A.; WILLS, G. Application of bagging, boosting and stacking to intrusion detection. In: PERNER, P. (Ed.). *Machine Learning and Data Mining in Pattern Recognition*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 593–602. ISBN 978-3-642-31537-4.

THORNE, J.; CHEN, M.; MYRIANTHOUS, G.; PU, J.; WANG, X.; VLACHOS, A. Fake news stance detection using stacked ensemble of classifiers. In: POPESCU, O.; STRAPPARAVA, C. (Ed.). *NLPmJ@EMNLP*. [S.l.]: Association for Computational Linguistics, 2017. p. 80–83. ISBN 978-1-945626-88-3.

THORNE, J.; VLACHOS, A.; CHRISTODOULOPOULOS, C.; MITTAL, A. FEVER: a large-scale dataset for fact extraction and verification. In: *NAACL-HLT*. [S.l.: s.n.], 2018.

Tie-Gang Fan; Ying Zhu; Jun-Min Chen. A new measure of classifier diversity in multiple classifier system. In: *2008 International Conference on Machine Learning and Cybernetics*. [S.l.: s.n.], 2008. v. 1, p. 18–21.

Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 20, n. 8, p. 832–844, 1998.

Tin Kam Ho; Hull, J. J.; Srihari, S. N. Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 16, n. 1, p. 66–75, Jan 1994.

VERMA, P.; SALISBURY, K. Unsupervised learning of audio perception for robotics applications: Learning to project data to t-sne/umap space. *CoRR*, abs/2002.04076, 2020. Disponível em: <http://dblp.uni-trier.de/db/journals/corr/corr2002.html#abs-2002-04076>.

VIJAYARAGHAVAN, S.; WANG, Y.; GUO, Z.; VOONG, J.; XU, W.; NASSERI, A.; CAI, J.; LI, L.; VUONG, K.; WADHWA, E. Fake news detection with different models. *CoRR*, abs/2003.04978, 2020.

VLACHOS, A. Automated fact checking. In: CUZZOCREA, A.; BONCHI, F.; GUNOPULOS, D. (Ed.). *CIKM Workshops*. [S.l.]: CEUR-WS.org, 2018. (CEUR Workshop Proceedings, v. 2482).

VOLKOVA, S.; SHAFFER, K.; JANG, J. Y.; HODAS, N. Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vancouver, Canada: Association for Computational Linguistics, 2017. p. 647–653.

VOSOUGHI, S.; ROY, D.; ARAL, S. The spread of true and false news online. *Science*, American Association for the Advancement of Science, v. 359, n. 6380, p. 1146–1151, 2018. ISSN 0036-8075.

WANG, W. Y. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In: BARZILAY, R.; KAN, M.-Y. (Ed.). *ACL (2)*. [S.l.]: Association for Computational Linguistics, 2017. p. 422–426. ISBN 978-1-945626-76-0.

WILBUR, W. J.; SIROTKIN, K. The automatic identification of stop words. *Journal of Information Science*, v. 18, n. 1, p. 45–55, 1992.

WOLD, S.; ESBENSEN, K.; GELADI, P. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, v. 2, n. 1, p. 37 – 52, 1987. ISSN 0169-7439. Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists. Disponível em: <http://www.sciencedirect.com/science/article/pii/0169743987800849>.

XU, S.; LI, Y.; WANG, Z. Bayesian multinomial naïve bayes classifier to text classification. In: PARK, J. J. J. H.; CHEN, S.-C.; CHOO, K.-K. R. (Ed.). *Advanced Multimedia and Ubiquitous Engineering*. Singapore: Springer Singapore, 2017. p. 347–352. ISBN 978-981-10-5041-1.

YULE, G. U. On the association of attributes in statistics: With illustrations from the material of the childhood society, c. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, The Royal Society, v. 194, p. 257–319, 1900. ISSN 02643952. Disponível em: <http://www.jstor.org/stable/90759>.

ZANNETTOU, S.; SIRIVIANOS, M.; BLACKBURN, J.; KOURTELLIS, N. The web of false information: Rumors, fake news, hoaxes, clickbait, and various other shenanigans. *J. Data and Information Quality*, ACM, New York, NY, USA, v. 11, n. 3, p. 10:1–10:37, maio 2019. ISSN 1936-1955.

ZHOU, Z.-H. *Ensemble Methods: Foundations and Algorithms*. 1st. ed. [S.l.]: Chapman Hall/CRC, 2012. ISBN 1439830037.

APÊNDICE A – RANGE DE PARÂMESTROS E MELHORES VALORES ENCONTRADOS EXPERIMENTO LIAR BINÁRIO

Técnica	Parametros/Range	Valor
ADB	n_estimators: 10, 50, 100, 120	CV=100 TFIDF: 100 Glove: 100 FastText: 100 W2V: 100
BNB	alpha: 0.0, 0.5, 1.0, 1.5, 1.7	CV=1.5 TFIDF=1.5 Glove=1.5 FastText=1.5 W2V=1.5
	binarize: 0.0, 0.5, 1.0, 1.5, 1.7	CV=0.0 TFIDF=0.0 Glove=0.0 FastText=0.0 W2V=0.0
CNB	alpha: 0.5, 1.0, 3.0, 5.5 , 6.0	CV=1.5 TFIDF=1.5 Glove=1.5 FastText=1.5 W2V=1.5
EXT	n_estimators: 40, 50, 150, 200	CV=150 TFIDF=150 Glove=150 FastText=150 W2V=150

Continua na próxima página

Técnica	**Parametros/Range**	**Valor**
GNB	var_smoothing 1e-5 1e-9 1e-10	CV=1e-9 TFIDF=1e-9 Glove=1e-9 FastText=1e-9 W2V=1e-9
KNN	n_neighbors: 3, 5, 10, 15, 20	CV=15 TFIDF=25 Glove=25 FastText=25 W2V=25
LR	penalty : L1, L2	CV=L1 TFIDF=L2 Glove=L2 FastText=L2 W2V=L2
	C:0.5, 1.0, 20.0, 25.0	CV=1.0 TFIDF=1.0 Glove=1.0 FastText=20.0 W2V=1.0
MLP	activation: logistic, relu	CV=relu TFIDF=relu Glove=relu FastText=relu W2V=relu

Continua na próxima página

Técnica	Parametros/Range	Valor
MNB	alpha : 0, 1, 4, 10, 11	CV=10 TFIDF=4 Glove=0 FastText=0 W2V=0
	fit_prior : True, False	CV=False TFIDF=False Glove=False FastText=True W2V=False
PA	C : 0.001, 0.01, 2.0, 2.5	CV=2 TFIDF=1 Glove=1 FastText=1 W2V=1
P	penalty: L1, L2, Elasticnet	CV=L1 TFIDF=L1 Glove=L1 FastText=L1 W2V=L1
	alpha: 1e-6, 1e-4, 0.1	CV=1e-4 TFIDF=1e-6 Glove=1e-6 FastText=1e-6 W2V=1e-6

Continua na próxima página

Técnica	Parametros/Range	Valor
RF	n_estimators: 5, 7, 30, 35, 100, 150, 170	CV=35 TFIDF=150 Glove=7 FastText=150 W2V=150
	criterion: entropy, gini	CV=gini TFIDF=gini Glove=gini FastText=gini W2V=gini
SGD	penalty: L1, L2	CV=L2 TFIDF=L2 Glove=L2 FastText=L2 W2V=L2
	alpha: 1e-2, 1e-4, 0.1	CV=1e-4 TFIDF=1e-4 Glove=1e-4 FastText=1e-4 W2V=1e-4
SVM	C: 0.5, 1.0, 1.5, 2.0	CV=1.0 TFIDF=1.0 Glove=1.0 FastText=1.0 W2V=1.0

Continua na próxima página

Técnica	**Parametros/Range**	**Valor**
	dual: True, Fals	CV=True
		TFIDF=True
		Glove=True
		FastText=True
		W2V=True

Tabela 19 – Range de parâmestros e melhores valores encontrados para os experimentos realizados na Seção 5.4 do Capítulo 5

APÊNDICE B – RANGE DE PARÂMESTROS E MELHORES VALORES ENCONTRADOS EXPERIMENTO LIAR 6 CLASSES

Técnica	Parametros/Range	Valor
ADB	n_estimators: 10, 50, 100, 120	CV=100 TFIDF: 100 Glove: 100 FastText: 100 W2V: 100
BNB	alpha: 0.0, 0.5, 1.0, 1.5, 1.7	CV=1.5 TFIDF=1.5 Glove=1.5 FastText=1.5 W2V=1.5
	binarize: 0.0, 0.5, 1.0, 1.5, 1.7	CV=0.0 TFIDF=0.0 Glove=0.0 FastText=0.0 W2V=0.0
CNB	alpha: 0.5, 1.0, 3.0, 5.5 , 6.0	CV=1.5 TFIDF=1.5 Glove=1.5 FastText=1.5 W2V=1.5
EXT	n_estimators: 40, 50, 150, 200	CV=150 TFIDF=150 Glove=150 FastText=150 W2V=150

Continua na próxima página

Técnica	**Parametros/Range**	**Valor**
GNB	var_smoothing 1e-5 1e-9 1e-10	CV=1e-9 TFIDF=1e-9 Glove=1e-9 FastText=1e-9 W2V=1e-9
KNN	n_neighbors: 3, 5, 10, 15, 20	CV=15 TFIDF=25 Glove=25 FastText=25 W2V=25
LR	penalty : L1, L2	CV=L1 TFIDF=L2 Glove=L2 FastText=L2 W2V=L2
	C:0.5, 1.0, 20.0, 25.0	CV=1.0 TFIDF=1.0 Glove=1.0 FastText=20.0 W2V=1.0
MLP	activation: logistic, relu	CV=relu TFIDF=relu Glove=relu FastText=relu W2V=relu

Continua na próxima página

Técnica	Parametros/Range	Valor
MNB	alpha : 0, 1, 4, 10, 11	CV=10 TFIDF=4 Glove=0 FastText=0 W2V=0
	fit_prior : True, False	CV=False TFIDF=False Glove=False FastText=True W2V=False
PA	C : 0.001, 0.01, 2.0, 2.5	CV=2 TFIDF=1 Glove=1 FastText=1 W2V=1
P	penalty: L1, L2, Elasticnet	CV=L1 TFIDF=L1 Glove=L1 FastText=L1 W2V=L1
	alpha: 1e-6, 1e-4, 0.1	CV=1e-4 TFIDF=1e-6 Glove=1e-6 FastText=1e-6 W2V=1e-6

Continua na próxima página

Técnica	Parametros/Range	Valor
RF	n_estimators: 5, 7, 30, 35, 100, 150, 170	CV=35
		TFIDF=150
		Glove=7
		FastText=150
		W2V=150
	criterion: entropy, gini	CV=gini
		TFIDF=gini
		Glove=gini
		FastText=gini
		W2V=gini
SGD	penalty: L1, L2	CV=L2
		TFIDF=L2
		Glove=L2
		FastText=L2
		W2V=L2
	alpha: 1e-2, 1e-4, 0.1	CV=1e-4
		TFIDF=1e-4
		Glove=1e-4
		FastText=1e-4
		W2V=1e-4
SVM	C: 0.5, 1.0, 1.5, 2.0	CV=1.0
		TFIDF=1.0
		Glove=1.0
		FastText=1.0
		W2V=1.0

Continua na próxima página

Técnica	**Parametros/Range**	**Valor**
	dual: True, Fals	CV=True
		TFIDF=True
		Glove=True
		FastText=True
		W2V=True

Tabela 20 – Range de parâmestros e melhores valores encontrados para os experimentos realizados na Seção 5.5 do Capítulo 5

APÊNDICE C – RANGE DE PARÂMESTROS E MELHORES VALORES ENCONTRADOS EXPERIMENTO FA-KES DATASE

Técnica	Parametros/Range	Valor
ADB	n_estimators: 10, 50, 100, 120	CV=100 TFIDF: 100 Glove: 100 FastText: 100 W2V: 100
BNB	alpha: 0.0, 0.5, 1.0, 1.5, 1.7	CV=1.5 TFIDF=1.5 Glove=1.5 FastText=1.5 W2V=1.5
	binarize: 0.0, 0.5, 1.0, 1.5, 1.7	CV=0.0 TFIDF=0.0 Glove=0.0 FastText=0.0 W2V=0.0
CNB	alpha: 0.5, 1.0, 3.0, 5.5 , 6.0	CV=1.5 TFIDF=1.5 Glove=1.5 FastText=1.5 W2V=1.5
EXT	n_estimators: 40, 50, 150, 200	CV=150 TFIDF=150 Glove=150 FastText=150 W2V=150

Continua na próxima página

Técnica	Parametros/Range	Valor
GNB	var_smoothing 1e-5 1e-9 1e-10	CV=1e-9 TFIDF=1e-9 Glove=1e-9 FastText=1e-9 W2V=1e-9
KNN	n_neighbors: 3, 5, 10, 15, 20	CV=15 TFIDF=25 Glove=25 FastText=25 W2V=25
LR	penalty : L1, L2	CV=L1 TFIDF=L2 Glove=L2 FastText=L2 W2V=L2
	C:0.5, 1.0, 20.0, 25.0	CV=1.0 TFIDF=1.0 Glove=1.0 FastText=20.0 W2V=1.0
MLP	activation: logistic, relu	CV=relu TFIDF=relu Glove=relu FastText=relu W2V=relu

Continua na próxima página

Técnica	Parametros/Range	Valor
MNB	alpha : 0, 1, 4, 10, 11	CV=10 TFIDF=4 Glove=0 FastText=0 W2V=0
	fit_prior : True, False	CV=False TFIDF=False Glove=False FastText=True W2V=False
PA	C : 0.001, 0.01, 2.0, 2.5	CV=2 TFIDF=1 Glove=1 FastText=1 W2V=1
P	penalty: L1, L2, Elasticnet	CV=L1 TFIDF=L1 Glove=L1 FastText=L1 W2V=L1
	alpha: 1e-6, 1e-4, 0.1	CV=1e-4 TFIDF=1e-6 Glove=1e-6 FastText=1e-6 W2V=1e-6

Continua na próxima página

Técnica	Parametros/Range	Valor
RF	n_estimators: 5, 7, 30, 35, 100, 150, 170	CV=35
		TFIDF=150
		Glove=7
		FastText=150
		W2V=150
	criterion: entropy, gini	CV=gini
		TFIDF=gini
		Glove=gini
		FastText=gini
		W2V=gini
SGD	penalty: L1, L2	CV=L2
		TFIDF=L2
		Glove=L2
		FastText=L2
		W2V=L2
	alpha: 1e-2, 1e-4, 0.1	CV=1e-4
		TFIDF=1e-4
		Glove=1e-4
		FastText=1e-4
		W2V=1e-4
SVM	C: 0.5, 1.0, 1.5, 2.0	CV=1.0
		TFIDF=1.0
		Glove=1.0
		FastText=1.0
		W2V=1.0

Continua na próxima página

Técnica	**Parametros/Range**	**Valor**
	dual: True, Fals	CV=True
		TFIDF=True
		Glove=True
		FastText=True
		W2V=True

Tabela 21 – Range de parâmestros e melhores valores encontrados para os experimentos realizados na Seção 5.6 do Capítulo 5